

**STATA TREATMENT-EFFECTS  
REFERENCE MANUAL:  
POTENTIAL OUTCOMES/COUNTERFACTUAL OUTCOMES  
RELEASE 17**



A Stata Press Publication  
StataCorp LLC  
College Station, Texas



Copyright © 1985–2021 StataCorp LLC  
All rights reserved  
Version 17

Published by Stata Press, 4905 Lakeway Drive, College Station, Texas 77845  
Typeset in  $\text{\TeX}$

ISBN-10: 1-59718-351-2

ISBN-13: 978-1-59718-351-2

This manual is protected by copyright. All rights are reserved. No part of this manual may be reproduced, stored in a retrieval system, or transcribed, in any form or by any means—electronic, mechanical, photocopy, recording, or otherwise—without the prior written permission of StataCorp LLC unless permitted subject to the terms and conditions of a license granted to you by StataCorp LLC to use the software and documentation. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document.

StataCorp provides this manual “as is” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. StataCorp may make improvements and/or changes in the product(s) and the program(s) described in this manual at any time and without notice.

The software described in this manual is furnished under a license agreement or nondisclosure agreement. The software may be copied only in accordance with the terms of the agreement. It is against the law to copy the software onto DVD, CD, disk, diskette, tape, or any other medium for any purpose other than backup or archival purposes.

The automobile dataset appearing on the accompanying media is Copyright © 1979 by Consumers Union of U.S., Inc., Yonkers, NY 10703-1057 and is reproduced by permission from CONSUMER REPORTS, April 1979.

Stata, **STATA** Stata Press, Mata, **MATA** and NetCourse are registered trademarks of StataCorp LLC.

Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations.

NetCourseNow is a trademark of StataCorp LLC.

Other brand and product names are registered trademarks or trademarks of their respective companies.

For copyright information about the software, type `help copyright` within Stata.

The suggested citation for this software is

StataCorp. 2021. *Stata: Release 17*. Statistical Software. College Station, TX: StataCorp LLC.

# Contents

Intro	Introduction to treatment-effects manual	1
Treatment effects	Introduction to treatment-effects commands	2
DID intro	Introduction to difference-in-differences estimation	5
didregress	Difference-in-differences estimation	14
didregress postestimation	Postestimation tools for didregress and xtdidregress	49
eteffects	Endogenous treatment-effects estimation	57
eteffects postestimation	Postestimation tools for eteffects	67
etpoisson	Poisson regression with endogenous treatment effects	72
etpoisson postestimation	Postestimation tools for etpoisson	85
etregress	Linear regression with endogenous treatment effects	89
etregress postestimation	Postestimation tools for etregress	117
stteffects	Treatment-effects estimation for observational survival-time data	122
stteffects intro	Introduction to treatment effects for observational survival-time data	123
stteffects ipw	Survival-time inverse-probability weighting	141
stteffects ipwra	Survival-time inverse-probability-weighted regression adjustment	151
stteffects postestimation	Postestimation tools for stteffects	179
stteffects ra	Survival-time regression adjustment	189
stteffects wra	Survival-time weighted regression adjustment	200
tebalance	Check balance after teffects or stteffects estimation	208
tebalance box	Covariate balance box	221
tebalance density	Covariate balance density	226
tebalance overid	Test for covariate balance	231
tebalance summarize	Covariate-balance summary statistics	236
teffects	Treatment-effects estimation for observational data	242
teffects intro	Introduction to treatment effects for observational data	243
teffects intro advanced	Advanced introduction to treatment effects for observational data	256
teffects aipw	Augmented inverse-probability weighting	269
teffects ipw	Inverse-probability weighting	291
teffects ipwra	Inverse-probability-weighted regression adjustment	299
teffects multivalued	Multivalued treatment effects	309
teffects nnmatch	Nearest-neighbor matching	318
teffects postestimation	Postestimation tools for teffects	332
teffects psmatch	Propensity-score matching	340
teffects ra	Regression adjustment	349
telasso	Treatment-effects estimation using lasso	361
telasso postestimation	Postestimation tools for telasso	383
teoverlap	Overlap plots	386
Glossary		394
Subject and author index		399

# Cross-referencing the documentation

When reading this manual, you will find references to other Stata manuals, for example, [U] [27 Overview of Stata estimation commands](#); [R] [regress](#); and [D] [reshape](#). The first example is a reference to chapter 27, *Overview of Stata estimation commands*, in the *User's Guide*; the second is a reference to the [regress](#) entry in the *Base Reference Manual*; and the third is a reference to the [reshape](#) entry in the *Data Management Reference Manual*.

All the manuals in the Stata Documentation have a shorthand notation:

[GSM]	<i>Getting Started with Stata for Mac</i>
[GSU]	<i>Getting Started with Stata for Unix</i>
[GSW]	<i>Getting Started with Stata for Windows</i>
[U]	<i>Stata User's Guide</i>
[R]	<i>Stata Base Reference Manual</i>
[BAYES]	<i>Stata Bayesian Analysis Reference Manual</i>
[CM]	<i>Stata Choice Models Reference Manual</i>
[D]	<i>Stata Data Management Reference Manual</i>
[DSGE]	<i>Stata Dynamic Stochastic General Equilibrium Models Reference Manual</i>
[ERM]	<i>Stata Extended Regression Models Reference Manual</i>
[FMM]	<i>Stata Finite Mixture Models Reference Manual</i>
[FN]	<i>Stata Functions Reference Manual</i>
[G]	<i>Stata Graphics Reference Manual</i>
[IRT]	<i>Stata Item Response Theory Reference Manual</i>
[LASSO]	<i>Stata Lasso Reference Manual</i>
[XT]	<i>Stata Longitudinal-Data/Panel-Data Reference Manual</i>
[META]	<i>Stata Meta-Analysis Reference Manual</i>
[ME]	<i>Stata Multilevel Mixed-Effects Reference Manual</i>
[MI]	<i>Stata Multiple-Imputation Reference Manual</i>
[MV]	<i>Stata Multivariate Statistics Reference Manual</i>
[PSS]	<i>Stata Power, Precision, and Sample-Size Reference Manual</i>
[P]	<i>Stata Programming Reference Manual</i>
[RPT]	<i>Stata Reporting Reference Manual</i>
[SP]	<i>Stata Spatial Autoregressive Models Reference Manual</i>
[SEM]	<i>Stata Structural Equation Modeling Reference Manual</i>
[SVY]	<i>Stata Survey Data Reference Manual</i>
[ST]	<i>Stata Survival Analysis Reference Manual</i>
[TABLES]	<i>Stata Customizable Tables and Collected Results Reference Manual</i>
[TS]	<i>Stata Time-Series Reference Manual</i>
[TE]	<i>Stata Treatment-Effects Reference Manual: Potential Outcomes/Counterfactual Outcomes</i>
[I]	<i>Stata Index</i>
[M]	<i>Mata Reference Manual</i>



# Title

**Intro** — Introduction to treatment-effects manual

[Description](#)   [Reference](#)   [Also see](#)

## Description

This manual documents commands for the analysis of treatment effects and is referred to as [TE] in cross-references.

After this entry, [TE] **Treatment effects** provides an overview of the treatment-effects estimation commands. The other parts of this manual are arranged alphabetically. If you are new to Stata's treatment-effects commands, we recommend that you read the following sections first:

[TE] <b>teffects intro</b>	Introduction to treatment effects for observational data
[TE] <b>teffects intro advanced</b>	Advanced introduction to treatment effects for observational data
[TE] <b>teffects multivalued</b>	Multivalued treatment effects

If you are interested in survival analysis, we also recommend that you read the following section first:

[TE] <b>stteffects intro</b>	Introduction to treatment effects for observational survival-time data
------------------------------	--

Stata is continually being updated, and Stata users are always writing new commands. To find out about the latest treatment-effects features, type `search treatment effects`.

## Reference

Andresen, M. E. 2018. *Exploring marginal treatment effects: Flexible estimation using Stata*. *Stata Journal* 18: 118–158.

## Also see

[U] **1.3 What's new**  
[R] **Intro** — Introduction to base reference manual

## Description

This manual documents commands that use observational data to estimate the effect caused by getting one treatment instead of another. In observational data, treatment assignment is not controlled by those who collect the data; thus some common variables affect treatment assignment and treatment-specific outcomes. Observational data is sometimes called retrospective data or nonexperimental data, but to avoid confusion, we will always use the term “observational data”.

When all the variables that affect both treatment assignment and outcomes are observable, the outcomes are said to be conditionally independent of the treatment, and the `teffects`, `stteffects`, `telasso`, `didregress`, and `xtdidregress` estimators may be used.

When not all of these variables common to both treatment assignment and outcomes are observable, the outcomes are not conditionally independent of the treatment, and `eteffects`, `etpoisson`, or `etregress` may be used.

`teffects` and `stteffects` offer much flexibility in estimators and functional forms for the treatment-assignment models. `teffects` provides models for continuous, binary, count, fractional, and nonnegative outcome variables. `stteffects` provides many functional forms for survival-time outcomes. See [\[TE\] teffects intro](#), [\[TE\] teffects intro advanced](#), and [\[TE\] stteffects intro](#) for more information. `telasso` provides models for continuous, binary, count, and nonnegative outcome variables, and allows for selection of covariates via lasso methods; see [\[TE\] telasso](#). `didregress` also provides models for continuous outcomes and is useful when treatment and control groups follow parallel trends before the treatment occurs. Beyond this, `xtdidregress` allows treatment-effect estimation when working with longitudinal or panel data. See [\[TE\] DID intro](#) and [\[TE\] didregress](#) for information on these commands.

`eteffects`, `etpoisson`, and `etregress` offer less flexibility than `teffects` because more structure must be imposed when conditional independence is not assumed. `eteffects` is for continuous, binary, count, fractional, and nonnegative outcomes and uses a probit model for binary treatments; see [\[TE\] eteffects](#). `etpoisson` is for count outcomes and uses a normal distribution to model treatment assignment; see [\[TE\] etpoisson](#). `etregress` is for linear outcomes and uses a normal distribution to model treatment assignment; see [\[TE\] etregress](#).

**Treatment effects**

[TE] <b>teffects aipw</b>	Augmented inverse-probability weighting
[TE] <b>teffects ipw</b>	Inverse-probability weighting
[TE] <b>teffects ipwra</b>	Inverse-probability-weighted regression adjustment
[TE] <b>teffects nnmatch</b>	Nearest-neighbor matching
[TE] <b>teffects psmatch</b>	Propensity-score matching
[TE] <b>teffects ra</b>	Regression adjustment
[TE] <b>telasso</b>	Treatment-effects estimation using lasso
[TE] <b>didregress</b>	Difference-in-differences estimation
[XT] <b>xtdidregress</b>	Fixed-effects difference-in-differences estimation

**Survival treatment effects**

[TE] <b>stteffects ipw</b>	Survival-time inverse-probability weighting
[TE] <b>stteffects ipwra</b>	Survival-time inverse-probability-weighted regression adjustment
[TE] <b>stteffects ra</b>	Survival-time regression adjustment
[TE] <b>stteffects wra</b>	Survival-time weighted regression adjustment

**Endogenous treatment effects**

[TE] <b>eteffects</b>	Endogenous treatment-effects estimation
[TE] <b>etpoisson</b>	Poisson regression with endogenous treatment effects
[TE] <b>etregress</b>	Linear regression with endogenous treatment effects

**Treatment effects with sample selection, endogenous covariates, and random effects**

[ERM] <b>eregress</b>	Extended linear regression
[ERM] <b>eintreg</b>	Extended interval regression
[ERM] <b>eprobit</b>	Extended probit regression
[ERM] <b>eoprobit</b>	Extended ordered probit regression

**Postestimation tools**

[TE] <b>tebalance</b>	Check balance after teffects or stteffects estimation
[TE] <b>tebalance box</b>	Covariate balance box
[TE] <b>tebalance density</b>	Covariate balance density
[TE] <b>tebalance overid</b>	Test for covariate balance
[TE] <b>tebalance summarize</b>	Covariate-balance summary statistics
[TE] <b>teoverlap</b>	Overlap plots
[TE] <b>eteffects postestimation</b>	Postestimation tools for eteffects
[TE] <b>etpoisson postestimation</b>	Postestimation tools for etpoisson
[TE] <b>etregress postestimation</b>	Postestimation tools for etregress
[TE] <b>stteffects postestimation</b>	Postestimation tools for stteffects
[TE] <b>telasso postestimation</b>	Postestimation tools for telasso
[TE] <b>didregress postestimation</b>	Postestimation tools for didregress and xtdidregress

## Also see

[U] [1.3 What's new](#)

[TE] [teffects intro](#) — Introduction to treatment effects for observational data

[TE] [teffects intro advanced](#) — Advanced introduction to treatment effects for observational data

[TE] [teffects multivalued](#) — Multivalued treatment effects

[TE] [stteffects intro](#) — Introduction to treatment effects for observational survival-time data

[TE] [Glossary](#)

# Title

**DID intro** — Introduction to difference-in-differences estimation

[Description](#)

[Remarks and examples](#)

[References](#)

[Also see](#)

## Description

Difference in differences (DID) is a method to estimate the average effect of a treatment on those who receive the treatment. The method can be applied to two types of observational data: repeated cross-sections, in which different individuals are observed at different time points, and panel data. As such, Stata provides two commands, `didregress` and `xtdidregress`. These commands estimate the average treatment effect on the treated (ATE) of a binary or continuous treatment on a continuous outcome by fitting a linear model with time fixed effects and group or panel fixed effects.

Difference-in-difference-in-differences (DDD) estimation is also available for situations in which you want to augment the DID framework to include additional control groups to obtain the ATE.

## Remarks and examples

Remarks are presented under the following headings:

*[Introduction](#)*

*[Intuition for estimating effects](#)*

*[Standard error considerations](#)*

*[Different types of data and specification](#)*

*[Specifying groups and time as binary indicators](#)*

*[Excluding group and time effects](#)*

*[Conclusion](#)*

This entry presents the intuition and some of the technical details for the estimators in `didregress` and `xtdidregress` and the diagnostics available after estimation. See [\[TE\] `didregress`](#) and [\[TE\] `didregress postestimation`](#) for details on the syntax and worked examples. For a more complete discussion and references on the subject, see [Angrist and Pischke \(2009\)](#), [Blundell and Dias \(2009\)](#), [Imbens and Wooldridge \(2009\)](#), [Lechner \(2011\)](#), [Angrist and Pischke \(2015\)](#), [Abadie and Cattaneo \(2018\)](#), and [Wing, Simon, and Bello-Gomez \(2018\)](#).

## Introduction

DID is one of the most venerable causal inference methods used by researchers. DID estimates the effect of a treatment on the treated group. When thinking about such effects, one usually thinks of comparing a group before and after the treatment, perhaps by looking at a graph such as the one below:



Figure 1

A treatment occurred in the year 2010. This might be a government policy, a change in medicine dosage, or any other intervention of interest. We would like to know if the treatment had an effect. It is clear from the graph that the outcome of interest changed after 2010. Is this due to the treatment or is something else occurring? Perhaps there are unobserved time effects that affect the treatment group after the treatment. For instance, there could have been a change in weather conditions or an economic downfall that affected the treatment group but was not captured in the model or the data. If this is the case, it does not suffice to look at the treatment group before and after the policy. DID addresses this by finding a control group, that is, a group that would have been subject to similar unobserved time effects and was not exposed to the treatment. Comparing the treatment group with the control group before and after the treatment might give us a better understanding of whether the treatment made a difference. A graph looking at a treatment and control group might look like

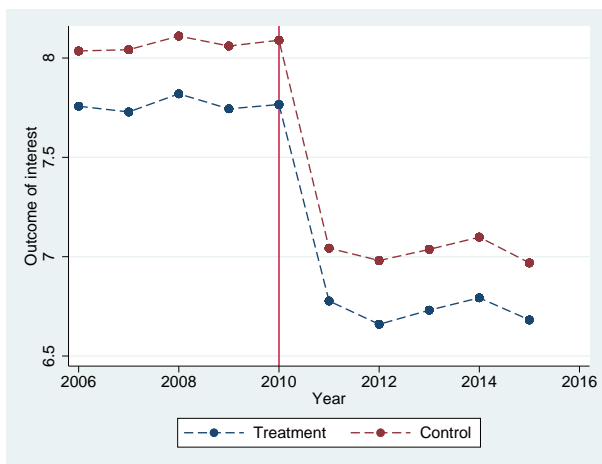


Figure 2

For both the treatment and the control group, we see that there was a decrease in the mean of the outcome after 2010. Therefore, the decrease we saw in the treatment group cannot be attributable entirely to the treatment. (In fact, these are simulated data, and we know the treatment has no effect.)

In a DID setup, if the treated group had not received the treatment, we would expect the treated and control groups to experience the same trends. A treatment effect would imply a systematic deviation from a common trend that could be observed graphically. This is what we observe:

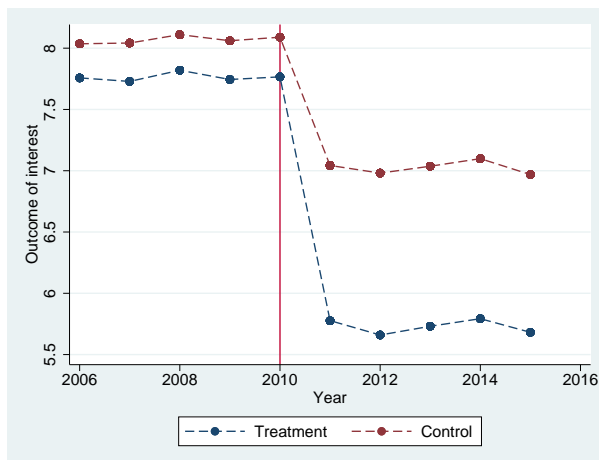


Figure 3

Here both groups experienced a decrease after 2010, but the treatment-group decrease was more substantial. The difference in the decreases across groups may indicate the effect of the treatment.

Researchers may motivate their analysis with such graphs. However, graphical evidence is not enough. We need statistical validation, so we fit a model.

The DID strategy relies on two differences. The first is a difference across time periods. Separately for the treatment group and the control group, we compute the difference of the outcome mean before and after the treatment. This across-time difference eliminates time-invariant unobserved group characteristics that confound the effect of the treatment on the treated group. But eliminating group-invariant unobserved characteristics is not enough to identify an effect. There may be time-varying unobserved confounders with an effect on the outcome mean, even after we control for time-invariant unobserved group characteristics. Therefore, we incorporate a second difference—a difference between the treatment group and the control group. DID eliminates time-varying confounders by comparing the treatment group with a control group that is subject to the same time-varying confounders as the treatment group.

The ATET is then consistently estimated by differencing the mean outcome for the treatment and control groups over time to eliminate time-invariant unobserved characteristics and also differencing the mean outcome of these groups to eliminate time-varying unobserved effects common to both groups. These two differences give the method its name and highlight its intuitive appeal. More appealing is the fact that you can get the effect of interest, the ATET, from one parameter in a linear regression.

When talking about DID, people cite [Snow \(1849\)](#) and [Snow \(1855\)](#) as the first known applications. Snow claimed that cholera was not transmitted by contaminated air or contaminated blood as was thought by some academics of his time. Snow hypothesized the disease was communicated via water that had been polluted with sewage. Below, he describes how he came up with an idea for a natural experiment to validate his hypothesis:

“In Thomas Street, Horsleydown, there are two courts close together, consisting of a number of small houses or cottages, inhabited by poor people. The houses occupy one side of each court or alley—the south side of Trusscott’s Court, and the north

side of the other, which is called Surrey Buildings, being placed back to back, with an intervening space, divided into small back areas, in which are situated the privies of both the courts, communicating with the same drain, and there is an open sewer which passes the further end of both courts. Now, in Surrey's buildings the cholera has committed fearful devastation, whilst in the adjoining court there has been but one fatal case, and another case that ended in recovery. In the former court the slops of dirty water poured down by the inhabitants into a channel in front of the houses got into the well from which they obtained their water, this being the only difference . . .”

In the first edition (1849) of the text, Snow reports the deaths from cholera from September 23, 1848 to August 25, 1849 for five London districts. The number of deaths is higher in the South and East districts relative to the other three districts, arising from the source of their water supply. Snow obtains a clear motivation for his theory. In the second edition (1855), he collects data before and after a pump with contaminated water in Broadstreet, London, is closed. It is then that he can compare a treated with a control group before and after a treatment to establish a treatment effect.

## Intuition for estimating effects

We can build our intuition about the causal inference implied by the DID setup by using the potential outcomes framework described in [TE] [teffects intro](#) and [TE] [teffects intro advanced](#). We consider individual-level data for which we sample different individuals at different points in time, a repeated cross-section. The treatment occurs at the group level. For example, the treatment may occur at the state, county, or hospital level. All individuals in a given state, county, or hospital either are treated or are controls at a given point in time. We index individuals by  $i$ , groups by  $g$ , and time by  $t$ . We are interested in the effect of a treatment,  $D_{igt} \in \{0, 1\}$ , on an outcome,  $Y_{igt}$ . Suppose the potential outcome mean of an individual in group  $g$  at time  $t$  that does not receive the treatment is given by the following:

$$E \{Y_{igt}(0) \mid g, t\} = \gamma_g + \gamma_t \quad (1)$$

Above,  $\gamma_g$  denotes the group effects and  $\gamma_t$  denotes the time effects. Also suppose the potential outcome mean for someone who receives the treatment is given by the following:

$$E \{Y_{igt}(1) \mid g, t\} = \gamma_g + \gamma_t + \delta \quad (2)$$

The potential outcome framework described above allows us to think of the regression model:

$$Y_{igt} = \gamma_g + \gamma_t + D_{gt}\delta + \varepsilon_{igt}$$

A regression estimate of  $\delta$ , the coefficient on the indicator of treatment, consistently estimates the ATET in this simplified framework, if we can make standard assumptions needed for linear regression and for treatment-effects estimators, discussed in [TE] [teffects intro advanced](#), plus one additional assumption.

To introduce this additional assumption, it is sometimes more intuitive to look at a two-period, two-group example. In this case,  $g \in \{0, 1\}$ , where 0 is the control group and 1 is the treatment group, and  $t \in \{0, 1\}$ , where 0 is the period before the treatment and 1 occurs after the treatment. To guarantee a consistent estimate of the ATET, we need to make the parallel-trends assumption:

$$E(Y_{i01} \mid D_{gt} = 1) - E(Y_{i00} \mid D_{gt} = 1) = E(Y_{i01} \mid D_{gt} = 0) - E(Y_{i00} \mid D_{gt} = 0)$$



The parallel-trends assumption is stated in terms of the potential outcomes of not being treated,  $Y_{i0t}$ , conditional on treatment,  $D_{gt}$ . It implies that if the treated had not received the treatment, the groups defined by  $D_{gt} = 1$  and  $D_{gt} = 0$  would have the same trends. For this to be true, we need group effects to be time invariant and time effects to be group invariant. The simple framework described in (1) and (2) satisfies the parallel-trends assumption.

The parallel-trends assumption has a graphical representation. Let's think again about the case with multiple time periods. The parallel-trends assumption implies the paths of the outcome variable over time are parallel between the control and treatment groups prior to the date of the treatment. We can visually check this assumption by plotting the means of the outcome over time for both groups or by visualizing the results of the linear trends model. For instance, we might use a graph like the one in [figure 2](#), where we plotted the means over time. After fitting a model using `didregress` and `xtdidregress`, you can get both the mean outcome plot and the trends plot by typing

```
. estat trendplots
```

Another way to think about the parallel-trends assumption in the pretreatment period is that treatment and control groups do not change their behavior in anticipation of the treatment. We can think of the parallel-trends assumption as implying that there should be no treatment effect in anticipation of the treatment. To test this assumption, we could fit a Granger-type causality model where we augment our model with dummies for each pretreatment–treatment period for the treated observations. A joint test of the coefficients on these dummies against 0 can be used as a test of the null hypothesis that no anticipatory effects have taken place. We can perform this test by typing

```
. estat granger
```

## Standard error considerations

While a standard linear regression model can be used to estimate the ATET, the best estimate of the standard error requires some consideration. Many standard error estimates have been proposed, and each one performs differently depending on the type of DID model being fit and the structure of the data. Here we provide a discussion of some of the issues centered on the available standard error estimates for `didregress` and `xtdidregress`. For a more complete discussion, see [Cameron and Miller \(2015\)](#) and [MacKinnon \(2019\)](#) and the references therein.

[Bertrand, Duflo, and Mullainathan \(2004\)](#) show that the standard errors for DID estimates are inconsistent if they do not account for the serial correlation of the outcome of interest. Because the outcomes studied usually vary at the group and time levels, it makes sense to correct for serial correlation. The authors show that using cluster–robust standard errors at the group level where treatment occurs provides correct coverage in the presence of serial correlation when the number of groups is not too small. [Bester, Conley, and Hansen \(2011\)](#) further show that using cluster–robust standard errors and using critical values of a  $t$  distribution with  $G - 1$  degrees of freedom, where  $G$  is the number of groups, is asymptotically valid for a fixed number of groups and a growing sample size. In other words, you do not need the number of groups to be arbitrarily large, that is, to grow asymptotically. Cluster–robust standard errors with  $G - 1$  degrees of freedom are the default standard errors of `didregress` and `xtdidregress`.

The results of [Bertrand, Duflo, and Mullainathan \(2004\)](#) and [Bester, Conley, and Hansen \(2011\)](#) demonstrate that we could still obtain reliable standard errors even when the number of groups was not large. But what about data with a very small number of groups? Several simulation and theoretical results suggest that cluster–robust standard errors may still have poor coverage when the number of groups is very small or when the number of treated groups is small relative to the number of control groups. These scenarios with small group sizes are common in DID studies, and another method of standard error estimation should be considered in these situations.

For cases where the number of groups is small, we provide three alternatives. The first alternative is to use the wild cluster bootstrap that imposes the null hypothesis that the ATET is 0. [Cameron, Gelbach, and Miller \(2008\)](#) and [MacKinnon and Webb \(2018\)](#) show that the wild cluster bootstrap provides better inference than using cluster-robust standard errors with  $t(G - 1)$  critical values. The second alternative comes from [Imbens and Kolesár \(2016\)](#), who show that with a small number of groups, you may use bias-corrected standard errors with the degrees of freedom adjustment proposed by [Bell and McCaffrey \(2002\)](#). For the third alternative, you may use aggregation type methods like those proposed by [Donald and Lang \(2007\)](#); they show that their method works well when the number of groups is small but the number of individuals in each group is large.

When the disparity between treatment and control groups is large, for example, because there is only one treated group or because the group sizes vary greatly, cluster-robust standard errors and the other methods mentioned above underperform. Yet the bias-corrected and cluster-bootstrap methods provide an improvement over the cluster-robust standard errors.

## Different types of data and specification

We allow for DID estimation of ATETs with two types of data: repeated cross-sections and panel data. The type of estimator to use depends on the assumptions you would like to make about the model and what type of data you have. Below, we walk you through the different types of models you may want to fit and how to fit them with `didregress` and `xtdidregress`.

Suppose you have a repeated cross-section of individuals  $i$  over a period of time  $t$ . The treatment is implemented at the state level,  $s$ , and it is denoted by  $D_{st}$ , where  $D_{st} = 1$  for all observations that are subject to the treatment in state  $s$  at time  $t$ . The indexing of the treatment makes clear that all individuals in the state at a given time are either treated or untreated. The model to be fit is given by

$$y_{ist} = \gamma_s + \gamma_t + \mathbf{z}_{ist}\beta + D_{st}\delta + \varepsilon_{ist} \quad (3)$$

In the expression above,  $y_{ist}$  is the outcome,  $y$  in our data;  $\mathbf{z}_{ist}$  are individual-level characteristics, given by `z1` and `z2`; and  $D_{st}$  is given by the variable `d`. To obtain the ATET, we type

```
. didregress (y z1 z2) (d), group(state) time(year)
```

Within the first set of parentheses, we list the outcome and covariates  $z_{ist}$  from (3). In the second set of parentheses, we specify the treatment variable. Group and time effects are included in the model through the `group()` and `time()` options, respectively. Also, by default, cluster-robust standard errors are computed at the `group()` level. The command above is equivalent to typing

```
. regress y z1 z2 i.year i.state d, vce(cluster state)
```

`didregress` also allows us to specify a DDD model for situations in which we would like to augment the DID framework to include another control group. For instance, starting from (3), let's assume the treatment occurs at the state level but also varies for people older than 65 versus people who are younger, defined by a binary variable  $g$ . The DDD model is now given by

$$y_{isgt} = \gamma_s + \gamma_g + \gamma_t + \gamma_s\gamma_t + \gamma_g\gamma_t + \gamma_s\gamma_g + \mathbf{z}_{isgt}\beta + D_{sgt}\delta + \varepsilon_{isgt}$$

To fit this model, we would type

```
. didregress (y z1 z2) (d), group(state g) time(year)
```

Although the model has a large set of interactions and looks much more complex than the DID model, the only difference in what we type is the addition of `g` to the set of group variables. The cluster-robust standard errors are computed at the highest level of clustering, in this case, `state`.

In some cases, data do not include a time component but rather multiple grouping variables across which differences may be taken. You could fit DDD models for three groups, say, `state`, `g1`, and `g2`, by typing

```
. didregress (y z1 z2) (d), group(state g1 g2)
```

Or, you could fit DID models without time by typing

```
. didregress (y z1 z2) (d), group(g1 g2)
```

With these last two specifications, diagnostics that rely on time variables are not available.

With panel data, the basic model is given by

$$y_{ist} = \gamma_i + \gamma_t + \mathbf{z}_{ist}\beta + D_{st}\delta + \varepsilon_{ist} \quad (4)$$

Again,  $i$  denotes the individual,  $t$  is the time period, and  $s$  is a group-level index. Individuals are assumed to be nested within the group, and treatment occurs at the group and time levels.

The specification is analogous to the one in (3) except that instead of  $\gamma_s$  we have  $\gamma_i$ . We assume the model in (4) is fit so that data comprising a panel of individuals is observed across the time for which individuals are nested within treatment group. Suppose that our data now include a panel identifier variable, `id`, which corresponds to  $i$  in (4). To fit the model above, we would type

```
. xtset id
. xtdidregress (y z1 z2) (d), group(state) time(year)
```

In this model, the group variable is different from the `xtset` identifying variable. The group variable defines the level of clustering, whereas the `xtset` variable defines the panel identifier. In other words, we are fitting a fixed-effects model with individual fixed-effects, not one with state fixed effects. Of course, the group and `xtset` variables could be the same, but they do not need to be.

## Specifying groups and time as binary indicators

The model discussed above is referred to as a generalized DID model by [Wing, Simon, and Bello-Gomez \(2018\)](#). It is a generalization of the two-period, two-group specification that is usually discussed in the literature. This is also the way [Angrist and Pischke \(2009\)](#) write the model.

It is not uncommon, however, to write the model in terms of binary indicators. Instead of having group effects and time effects in the model for all groups and times in the data, you could instead include only a posttreatment effect and a treatment-group effect. To do this, you would create and include an indicator in your model that takes the value of 1 if the observation belongs to a treated group and 0 otherwise as well as an indicator that is 1 if the period considered occurs after the treatment. You can specify this model by using the `nocteffects` option and including your own indicators.

Say, for example, you create a time-constant indicator, `gtreated`, that is 1 if a group is treated and 0 otherwise, as well as an indicator, `post`, that is 1 after the policy was implemented and 0 before. You would type

```
. didregress (y z1 z2 i.post i.gtreated) (d), group(state) time(year) nocteffects
```

The syntax is equivalent for `xtdidregress`.

The model you specify in this case is given by

$$y_{ist} = 1 \{s = \text{treated}\} + 1 \{t = \text{post}\} + \mathbf{z}_{ist}\beta + D_{st}\delta + \varepsilon_{ist}$$

where  $1 \{s = \text{treated}\}$  is an indicator function that is 1 if group  $s$  receives the treatment and 0 otherwise. Similarly,  $1 \{t = \text{post}\}$  indicates that we are in the period for which the treatment is active. In the expression above, we still specify the `group()` and `time()` options to obtain the correct standard errors, to validate the assumptions of the DID specification, and to obtain group- and time-level statistics.

## Excluding group and time effects

The DID commands also allow you to forgo the group and time effects that are generated by default. In this case, you would have to add your own group and time effects or omit them entirely by using the `noeffect` option, which excludes group and time effects from your specification. For DDD models, you may also use the `nointeract` option, which will exclude group and time interactions from your model. Combining both options excludes group and time effects altogether. You may type

```
. didregress (y z) (d), group(s g) time(t) noeffect nointeract
```

You would then fit

$$y_{isgt} = \beta_0 + \mathbf{z}_{igst}\beta + D_{sgt}\delta + \varepsilon_{ist}$$

You could still include group and time interactions by adding them in the first set of parentheses.

## Conclusion

`didregress` and `xtdidregress` compute the ATET using DID and DDD for panel data and repeated cross-sections. Both commands offer standard error computations that address some of the issues that researchers face, such as a small number of groups. After fitting models with these commands, diagnostics and tests to validate the assumptions and internal validity of the DID and DDD results are available.

## References

- Abadie, A., and M. D. Cattaneo. 2018. Econometric methods for program evaluation. *Annual Review of Economics* 10: 465–503. <https://doi.org/10.1146/annurev-economics-080217-053402>.
- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.
- . 2015. *Mastering Metrics: The Path from Cause to Effect*. Princeton, NJ: Princeton University Press.
- Bell, R. M., and D. F. McCaffrey. 2002. Bias reduction in standard errors for linear regression with multi-stage samples. *Survey Methodology* 28: 169–181.
- Bertrand, M., E. Duflo, and S. Mullainathan. 2004. How much should we trust difference-in-differences estimates? *Quarterly Journal of Economics* 119: 249–275. <https://doi.org/10.1162/003355304772839588>.
- Bester, C. A., T. G. Conley, and C. B. Hansen. 2011. Inference with dependent data using cluster covariance estimators. *Journal of Econometrics* 165: 137–151. <https://doi.org/10.1016/j.jeconom.2011.01.007>.
- Blundell, R. W., and M. C. Dias. 2009. Alternative approaches to evaluation in empirical microeconomics. *Journal of Human Resources* 44: 565–640. <https://doi.org/10.1007/s10258-002-0010-3>.

- Cameron, A. C., J. B. Gelbach, and D. L. Miller. 2008. Bootstrap-based improvements for inference with clustered errors. *Review of Economics and Statistics* 90: 414–427. <https://doi.org/10.1162/rest.90.3.414>.
- Cameron, A. C., and D. L. Miller. 2015. A practitioner’s guide to cluster-robust inference. *Journal of Human Resources* 50: 317–372. <https://doi.org/10.3368/jhr.50.2.317>.
- Donald, S. G., and K. Lang. 2007. Inference with difference-in-differences and other panel data. *Review of Economics and Statistics* 89: 221–233. <https://doi.org/10.1162/rest.89.2.221>.
- Imbens, G. W., and M. Kolesár. 2016. Robust standard errors in small samples: Some practical advice. *Review of Economics and Statistics* 98: 701–712. [https://doi.org/10.1162/REST\\_a\\_00552](https://doi.org/10.1162/REST_a_00552).
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.
- Kaspereit, T. 2021. [Event studies with daily stock returns in Stata: Which command to use?](#) *Stata Journal* 21: 462–497.
- Lechner, M. 2011. The estimation of causal effects by difference-in-difference methods. *Foundations and Trends in Econometrics* 4: 165–224. <http://doi.org/10.1561/08000000014>.
- MacKinnon, J. G. 2019. How cluster-robust inference is changing applied econometrics. *Canadian Journal of Economics* 52: 851–881. <https://doi.org/10.1111/caje.12388>.
- MacKinnon, J. G., and M. D. Webb. 2018. The wild bootstrap for few (treated) clusters. *Econometrics Journal* 21: 114–135. <https://doi.org/10.1111/ectj.12107>.
- Roodman, D., J. G. MacKinnon, M. O. Nielsen, and M. D. Webb. 2019. [Fast and wild: Bootstrap inference in Stata using boottest](#). *Stata Journal* 19: 4–60.
- Snow, J. 1849. *On the Mode of Communication of Cholera*. London: Churchill.
- . 1855. *On the Mode of Communication of Cholera*. 2nd ed. London: Churchill.
- Wing, C., K. Simon, and R. A. Bello-Gomez. 2018. Designing difference in difference studies: Best practices for public health policy research. *Annual Review of Public Health* 39: 453–469. <https://doi.org/10.1146/annurev-publhealth-040617-013507>.

## Also see

[TE] [teffects](#) — Treatment-effects estimation for observational data

[TE] [teffects intro](#) — Introduction to treatment effects for observational data

[TE] [teffects intro advanced](#) — Advanced introduction to treatment effects for observational data

[Description](#)[Quick start](#)[Menu](#)[Syntax](#)[Options](#)[Remarks and examples](#)[Stored results](#)[Methods and formulas](#)[Acknowledgment](#)[References](#)[Also see](#)

## Description

`didregress` estimates the average treatment effect on the treated (ATET) from observational data by difference-in-differences (DID) or difference-in-difference-in-differences (DDD). The ATET of a binary or continuous treatment on a continuous outcome is estimated by fitting a linear model with time and group fixed effects. The DID and DDD estimation performed by `didregress` can be applied to data comprising repeated cross-sections in which different groups of individuals are observed at each time period.

`xtdidregress` estimates the ATET from observational data by DID or DDD for panel data. The ATET of a binary or continuous treatment on a continuous outcome is estimated by fitting a linear model with time and panel fixed effects.

## Quick start

DID estimate of the ATET of `treat1` on outcome `y1` modeled using covariates `x1` and `x2`, and `grpvar1` and `tvar` fixed effects, with the treatment occurring at the `grpvar1` and `tvar` levels

```
didregress (y1 x1 x2) (treat1), group(grpvar1) time(tvar)
```

As above, but compute wild cluster–bootstrap *p*-values and confidence intervals with `grpvar1` as the clustering variable

```
didregress (y1 x1 x2) (treat1), group(grpvar1) time(tvar) ///
wildbootstrap
```

Aggregate data at the `grpvar1` and `tvar` levels to estimate the ATET

```
didregress (y1 x1 x2) (treat1), group(grpvar1) time(tvar) ///
aggregate(standard)
```

As above, but use the [Donald and Lang \(2007\)](#) method to compute the ATET and standard errors

```
didregress (y1 x1 x2) (treat1), group(grpvar1) time(tvar) ///
aggregate(dlang)
```

DDD estimate of the ATET of `treat2` on outcome `y2` modeled using covariates `x1` and `x2` and fixed effects defined by two-way interactions for `grpvar1`, `grpvar2`, and `tvar`, with the treatment occurring at the `grpvar1`, `grpvar2`, and `tvar` levels

```
didregress (y2 x1 x2) (treat2), group(grpvar1 grpvar2) time(tvar)
```

DID estimate of ATET of `treat3` on outcome `y3` using `xtset` data; `y3` modeled using covariates `x1` and `x2`, and individual (panel) and `tvar` fixed effects, with the treatment occurring at the `grpvar1` and `tvar` levels

```
xtdidregress (y3 x1 x2) (treat3), group(grpvar1) time(tvar)
```

## Menu

### didregress

Statistics > Treatment effects > Continuous outcomes > Difference in differences

### xtdidregress

Statistics > Treatment effects > Continuous outcomes > Difference in differences (FE)

Statistics > Longitudinal/panel data > Difference in differences

## Syntax

*DID with repeated cross-sectional data*

```
didregress (ovar omvarlist) (tvar [, continuous]) [if] [in] [weight],
           group(groupvars) [time(timevar) options]
```

*DID with longitudinal data*

```
xtdidregress (ovar omvarlist) (tvar [, continuous]) [if] [in] [weight],
             group(groupvars) [time(timevar) options]
```

*ovar* is the outcome of interest.

*omvarlist* specifies the covariates in the outcome model and may contain factor variables; see [\[U\] 11.4.3 Factor variables](#).

*tvar* must be a binary variable indicating observations subject to treatment or a continuous variable measuring treatment intensity.

*groupvars* are categorical variables that indicate the group level at which the treatment occurs. At least one group variable must be specified. If *timevar* is specified, at most two group variables may be specified. If *timevar* is not specified, at most three group variables may be specified.

*timevar* is a time variable. It must be specified if *groupvar* has only one variable.

<i>options</i>	Description
Model	
* <b>group</b> ( <i>groupvars</i> )	specify group variables
† <b>time</b> ( <i>timevar</i> )	specify time variable
<b>nointeract</b>	exclude <b>group()</b> and <b>time()</b> interactions
<b>nogteffects</b>	do not include group and time effects in the model
<b>aggregate</b> ( <i>aggmethod</i> )	aggregate to the levels defined by interacting <i>groupvars</i> and <i>timevar</i>
<b>wildbootstrap</b> [ ( <i>wildopts</i> ) ]	compute confidence intervals and <i>p</i> -values with the wild bootstrap
SE/Robust	
<b>vce</b> ( <i>vcetype</i> )	<i>vcetype</i> may be <b>cluster</b> <i>clustvar</i> , <b>robust</b> , <b>hc2</b> , or <b>bootstrap</b>
Reporting	
<b>level</b> (#)	set confidence level; default is <b>level</b> (95)
<b>aequations</b>	display auxiliary-equation results
<b>display_options</b>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
<b>coeflegend</b>	display legend instead of statistics

\* **group**(*groupvars*) is required.

† **time**(*timevar*) is required when only one group is specified.

**by**, **collect**, and **statsby** are allowed; see [U] 11.1.10 Prefix commands.

**fweights**, **awweights**, and **pweights** are allowed; see [U] 11.1.6 weight.

**coeflegend** does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

<i>aggmethod</i>	Description
<b>standard</b>	aggregate data and fit model
<b>dlang</b> [ , <i>dlopt</i> ]	aggregate data and fit model using Donald and Lang method

<i>wildopts</i>	Description
<b>errorweight</b> ( <i>edtype</i> )	specify the error weight type <i>edtype</i> ; default is <b>errorweight</b> ( <b>rademacher</b> )
<b>reps</b> (#)	perform # wild bootstrap replications; default is <b>reps</b> (1000)
<b>rseed</b> (#)	set random-number seed to #
<b>blocksize</b> (#)	perform wild bootstrap in blocks of # replications

## Options

### Model

**group**(*groupvars*) specifies group variables. It indicates the group level at which the treatment occurs. *groupvars* may be, for example, states, counties, or hospitals. *groupvars* define the group levels for which group effects are included in the model used to perform DID estimation and for which group interactions are included in the model used to perform DDD estimation. **group()** also defines the level clustering for the default cluster-robust standard errors. **group()** is required.



You may specify at most two group variables if you also specify a time variable in the `time()` option or at most three group variables if no time variable is specified.

`time(timevar)` specifies the time variable. You may specify `time()` when one or two group variables are specified in the `group()` option. `time()` is required when only one group variable is specified in the `group()` option.

`nointeract` excludes *groupvars* and *timevar* interactions from being included in the model. By default, `didregress` and `xtdidregress` include group and time interactions for your specification if there is more than one group variable.

`nogteffects` specifies to not include group and time effects. By default, `didregress` adds group and time dummies to the regression specification. By default, `xtdidregress` adds time dummies to the fixed-effects specification.

`aggregate(aggmethod)` fits the model by aggregating data at the *groupvars* and *timevar* levels. *aggmethod* may be either `standard` or `dlang[, dlopt]`.

`standard` specifies that aggregation is performed using the standard aggregation method. In this case, a regression model is fit of the original outcome on covariates that vary within levels of the *groupvars* and *timevar* interaction. The estimates of the group-time level effects from this regression are used to construct a new dependent variable. These effects along with the remaining covariates are then aggregated to the level of the of the *groupvars* and *timevar* interaction. The final results are obtained by regressing the estimated group-time level effects on the remaining covariates with this aggregated dataset and estimating group-level cluster-robust standard errors.

`dlang[, dlopt]` aggregates data using methods proposed by Donald and Lang (2007). *dlopt* may be either `constant` or `varying`.

`constant` requests that standard errors be estimated using the standard ordinary least-squares method, as suggested by Donald and Lang (2007). With this method, as with the `standard` aggregation method, a single regression model is fit in the first step, so the coefficient estimates are constant across levels of the *groupvars* and *timevar* interaction.

`varying` specifies that the aggregation method allow for varying coefficients on the covariates. Specifically, in this case, separate regression models of the original outcome on covariates that vary within levels of *groupvars* and *timevar* are fit for each level of the *groupvars* and *timevar* interaction. Thus, this method allows the coefficients on these variables to vary. The constant from each of these regressions forms the new dependent variable. The final regression and standard error computations are equivalent to those used by the `constant` method.

`wildbootstrap[ (wildopts) ]` computes confidence intervals and *p*-values with the wild bootstrap. The wild bootstrap is constructed imposing the null hypothesis that the ATET is 0; that is, it is a restricted wild bootstrap. Confidence intervals are computed separately from the *p*-values. The bounds of the confidence interval are computed using a bisection optimization algorithm described in *Methods and formulas*. *wildopts* are `errorweight(edtype)`, `reps(#)`, `rseed(#)`, and `blocksize(#)`.

`errorweight(edtype)` defines the error weight used to draw residuals from the wild bootstrap. *edtype* is one of `rademacher` (the default), `mammen`, `webb`, `normal`, or `gamma`.

`rademacher` multiplies the residuals at each bootstrap replication with a randomly generated variable that takes the value of 1 with probability 0.5 and the value of  $-1$  with probability 0.5. `errorweight(rademacher)` is the default.

`mammen` multiplies the residuals at each bootstrap replication with a randomly generated variable that takes the value of  $1 - \phi$  with probability  $\phi/\sqrt{5}$  and  $\phi$  otherwise, where  $\phi = (1 + \sqrt{5})/2$ .

`webb` multiplies the residuals at each bootstrap replication with a randomly generated variable that takes the values  $-\sqrt{3/2}$ ,  $-\sqrt{2/2}$ ,  $-\sqrt{1/2}$ ,  $\sqrt{1/2}$ ,  $\sqrt{2/2}$ , and  $\sqrt{3/2}$ , each with probability  $1/6$ .

`normal` multiplies the residuals at each bootstrap replication with a randomly generated normal distribution variable with the first four moments given by 0, 1, 0, and 3.

`gamma` multiplies the residuals at each bootstrap replication with a randomly generated gamma distribution variable with shape parameter 4 and scale parameter  $1/2$ .

`reps(#)` performs `#` wild bootstrap replications. The default is `reps(1000)`.

`rseed(#)` sets the random-number seed to `#`.

`blocksize(#)` specifies that the wild bootstrap be performed in blocks, with `#` replications per block. The wild bootstrap computation requires a matrix with dimensions  $(\# \text{ groups}) \times (\# \text{ replications})$ . If this is too large, you can reduce the matrix to  $(\# \text{ groups}) \times (\# \text{ block size})$  and loop  $(\# \text{ replications}) / (\# \text{ block size})$  times. When the same random seed is set, using a different block size does not change the numerical results; it only modifies the computation method. The block size must be less than or equal to the number of bootstrap replications.

#### SE/Robust

`vce(vctype)` specifies the type of standard error reported, which includes types that allow for intragroup correlation (`cluster clustvar`), that are robust to some kinds of misspecification (`robust`), that are bias-corrected cluster robust using the degrees-of-freedom adjustment proposed by Bell and McCaffrey (2002) (`hc2`), and that use bootstrap sampling done at the group level (`bootstrap`); see [R] [vce\\_option](#).

`vce(cluster clustvar)`, the default, uses the first variable specified in the `group(groupvars)` option.

`vce(hc2)` specifies bias-corrected cluster-robust standard errors with the degrees-of-freedom adjustment proposed by Bell and McCaffrey (2002). As with `vce(hc2)` in [R] [regress](#), the residuals are rescaled by the projection matrix to improve the small-sample properties of the variance estimates. For more details, see [Methods and formulas](#).

Specifying `vce(robust)` is equivalent to specifying `vce(cluster clustvar)`, where `clustvar` is the first variable specified in the `group(groupvars)` option.

#### Reporting

`level(#)`; see [R] [Estimation options](#).

`aequations` specifies that the results for the outcome-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

`display_options`: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

The following option is available with `didregress` and `xtdidregress` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

## Remarks and examples

Remarks are presented under the following headings:

*Introduction*  
*DID estimation*  
*Graphical diagnostics and tests*  
*Specifying a 2-by-2 DID*  
*Standard-error considerations*  
*Default cluster-robust standard errors*

### Introduction

DID is one of the most venerable causal inference methods used by researchers. DID estimates the average treatment effect on the treated group (ATET).

To obtain the ATET using DID, one must compute the difference of the mean outcome for the treatment and the control groups before and after the treatment. This eliminates time-invariant unobservable group characteristics that confound the ATET; however, this is not enough to identify an effect. There may be time-varying unobservable confounders with an effect on the treatment group even after we control for time-invariant unobservable group characteristics. DID eliminates time-varying confounders by including a control group that is subject to the same time-varying confounders as the treatment group.

The ATET is then consistently estimated, differencing the mean outcome for the treatment and control groups over time to eliminate time-invariant unobservable characteristics and also differencing the mean outcome of these groups to eliminate time-varying unobservable effects common to both groups. These two differences give the DID method its name and highlight its intuitive appeal. More appealing is the fact that you can get the effect of interest, the ATET, from one parameter in a linear regression.

Below, we illustrate how to use `didregress` and `xtdidregress`. For more information about the methods used below, see [TE] [DID intro](#). For general discussions of the DID methodology, see [Angrist and Pischke \(2009, 2015\)](#), [Blundell and Dias \(2009\)](#), [Imbens and Wooldridge \(2009\)](#), [Lechner \(2011\)](#), [Abadie and Cattaneo \(2018\)](#), and [Wing, Simon, and Bello-Gomez \(2018\)](#) and the references therein.

### DID estimation

#### ► Example 1: Fitting a DID model

A health provider is interested in studying the effect of a new hospital admissions procedure on the satisfaction of patients. The provider has monthly data on patients from January to July. The new admissions procedure was implemented in April by hospitals that were under new management. Of the 46 hospitals in the study, 18 implemented the new procedure.

The health provider will use a DID regression to analyze the effect of the new admissions procedure on the hospitals that participated in the program. The outcome of interest is patient satisfaction, `satis`, which is recorded as an average of the responses to a set of four questions asked to patients. `satis` may take values between 0 and 10, where 10 is the greatest possible level of satisfaction and 0 is utter disappointment. The `procedure` variable marks the treated observations; it is 1 if a surveyed individual was admitted to the hospital using the new procedure after March and 0 otherwise. To get the ATET on the outcome `satis`, we type

```
. didregress (satis) (procedure), group(hospital) time(month)
```

The first set of parentheses is used to specify the outcome of interest followed by the covariates in the model. In this case, there are no covariates, just the outcome, `satis`. The second set of parentheses is used to specify the binary variable that indicates the treated observations, `procedure`. The `group()` and `time()` options are used to construct group and time fixed effects that are included in the model. The variable specified in `group()` is also important because it defines the level of clustering for the default cluster-robust standard errors; in this case, we cluster at the hospital level. The results from this command are as follows:

```
. use https://www.stata-press.com/data/r17/hospdd
(Artificial hospital admission procedure data)
. didregress (satis) (procedure), group(hospital) time(month)
Number of groups and treatment time
Time variable: month
Control:      procedure = 0
Treatment:    procedure = 1
```

	Control	Treatment
Group hospital	28	18
Time Minimum	1	4
Maximum	1	4

```
Difference-in-differences regression          Number of obs = 7,368
Data type: Repeated cross-sectional
              (Std. err. adjusted for 46 clusters in hospital)
```

	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]
satis					
ATET procedure (New vs Old)	.8479879	.0321121	26.41	0.000	.7833108 .912665

Note: ATET estimate adjusted for group effects and time effects.

The first table gives information about the treated and control groups and about treatment timing. The first segment with the title `Group` tells us the number of treated and control hospitals: 28 hospitals were using the old procedure and 18 hospitals were using the new one. The second segment of the table gives information about the first time we observe hospitals in the control group and the first time we observe the treatment (the new admission procedure) for hospitals in the treatment group. In this example, all hospitals that adopted the new procedure did so in April, time period 4. If some hospitals had adopted the policy later, the minimum and maximum time of first treatment would differ.

The ATET is 0.85, almost a 1-point increase in satisfaction relative to the case where none of the treated hospitals enacted the new procedure. In other words, if the hospitals that implemented the new admission procedure had not done so, their satisfaction ratings would be lower by almost one point on average.

We now explore whether the trajectories of `satis` are parallel for the control and treatment groups prior to the date when the new procedure was implemented. We are checking what is known as the parallel-trends or common-trends assumption, an important assumption of the DID model, as discussed in [TE] **DID intro**. A visual diagnostic of this assumption can be obtained by plotting the means of

the outcome over time for both groups or by visualizing the results of the linear-trends model. We can perform both of these diagnostic checks by using `estat trendplots`. To obtain figure 1 below, we type

```
. estat trendplots
```

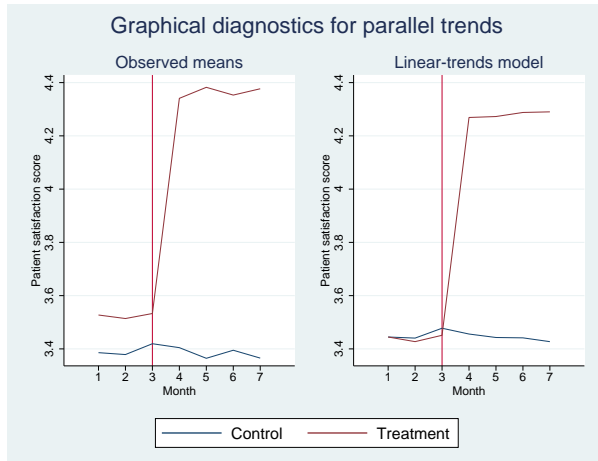


Figure 1

The graph seems to indicate that the parallel-trends assumption is satisfied. Prior to the policy implementation, treated and control hospitals followed a parallel path.

We could also perform a test to see if the trajectories are parallel by augmenting our original model to include variables representing time trends before and after the treatment for both groups of hospitals. The linear-trends model estimates a coefficient for the differences in linear trends prior to treatment, and if that coefficient is 0, the linear pretreatment trends are parallel. Otherwise, identification of the ATET may become questionable.

We can perform this test by using `estat ptrends`:

```
. estat ptrends
Parallel-trends test (pretreatment time period)
H0: Linear trends are parallel
F(1, 45) = 0.55
Prob > F = 0.4615
```

We do not have evidence to reject the null hypothesis of parallel trends in this case. Both the test and the graphical analysis support the parallel-trends assumption and, therefore, our ATET estimate. ◀

## ▶ Example 2: Fitting a DDD model

The results in [example 1](#) could come into question if they could be the consequence of other unobserved variables rather than the consequence of the new hospital admissions procedure. The health provider administrators believe that responses to the survey are related to the frequency of individuals' hospital visits. The patients may have unobserved characteristics that affect both how frequently they visit the hospital and how they feel about the admissions procedure. In other words, there might be unobserved characteristics that confound the effect of the new hospital admissions procedure. The administrators decide to obtain the ATET by using a DDD model. They want to estimate the average treatment effect on patients who visit the hospital with high or very high frequencies.

To do this, we will first create a new variable `hightrt` to be our new treatment identifier. Observations are now marked as treated (`hightrt = 1`) if hospital visit frequency by an individual is high or very high (`frequency = 3` or `4`) and if the hospital implemented the new admissions procedure in April.

```
. generate hightrt = procedure==1 & (frequency==3 | frequency==4)
. label define trt 0 "Untreated" 1 "Treated"
. label values hightrt trt
```

The DDD model will incorporate both hospital and frequency of usage effects as well as their interaction with time effects. To fit the model, we incorporate a new group variable, `frequency`:

```
. didregress (satis) (hightrt), group(hospital frequency) time(month)
(output omitted)
```

```
Number of groups and treatment time
Time variable: month
Control:      hightrt = 0
Treatment:    hightrt = 1
```

	Control	Treatment
Group		
hospital	28	18
frequency	2	2
Time		
Minimum	1	4
Maximum	1	4

```
Triple-differences regression Number of obs = 7,368
Data type: Repeated cross-sectional
(Std. err. adjusted for 46 clusters in hospital)
```

satis	Robust		t	P> t	[95% conf. interval]	
	Coefficient	std. err.				
ATET						
hightrt						
(Treated						
vs						
Untreated)	.764154	.0402603	18.98	0.000	.6830655	.8452425

Note: ATET estimate adjusted for group effects, time effects, and group- and time-effects interactions.

The omitted output after the command corresponds to the factor-variable interactions that include the base categories. This is common when you fit DDD models that by default include group interactions and group and time interactions.

The first table above has information on the second group variable, `frequency`, for which low and medium frequencies are controls and high and very high frequencies are treated. The second table shows that the ATET is now smaller, but the policies still increase satisfaction.

### ▷ Example 3: DID for panel-data model

Moser and Voena (2012) look at the effect of compulsory licensing on domestic inventions. Compulsory licensing allows firms in developing countries to produce foreign inventions without the consent of foreign patent owners. Having access to foreign technology may discourage domestic inventions, but it could also enhance local production.

Moser and Voena consider legislation that occurred during World War I called Trading With the Enemy Act (TWEA). By 1919, German-owned patents were systematically licensed to U.S. companies. In Moser and Voena (2012), the treated observations correspond to a subclass in the chemical industry that was granted at least one of the TWEA patents after 1918, as reported by United States Patent and Trademark Office (USPTO). A subclass is a group of firms in an industry that employ similar technologies, as defined by the USPTO.

The outcome of interest is the number of patents granted to inventors from the U.S. in that subclass, `uspatents`. This measures domestic innovation. Moser and Voena also include the number of non-TWEA patents granted to the subclass that were from foreign inventors, `fpatents`. `fpatents` measure innovation in the subclass that is not from U.S. inventors. We observe the same subclasses at each point in time from 1875 to 1939. Thus, we have a panel dataset.

To fit the model, we first `xtset` our data at the subclass level, `classid`:

```
. use https://www.stata-press.com/data/r17/patents
(Excerpt from Moser and Voena (2012))
. xtset classid
Panel variable: classid (balanced)
```

Below, we fit a DID model for the number of patents granted to U.S. inventors in a subclass, controlling for the number of non-TWEA patents granted to foreign inventors. The treatment indicator `gotpatent` is 1 if the subclass received a TWEA patent after 1918 and is 0 otherwise. The model includes year fixed effects and subclass fixed effects.

```
. xtdidregress (uspatents fpatents) (gotpatent), group(classid) time(year)
Number of groups and treatment time
Time variable: year
Control:      gotpatent = 0
Treatment:    gotpatent = 1
```

	Control	Treatment
Group		
classid	6912	336
Time		
Minimum	1875	1919
Maximum	1875	1919

Difference-in-differences regression Number of obs = 471,120  
 Data type: Longitudinal  
 (Std. err. adjusted for 7,248 clusters in classid)

uspatents	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]
ATET gotpatent (Patent vs None)	.150516	.0356081	4.23	0.000	.0807137 .2203183

Note: ATET estimate adjusted for covariates, panel effects, and time effects.

The ATET is 0.15, which means that in subclasses that were awarded one or more patents, domestic inventors produced an average of 0.15 additional patents after the TWEA compared with the scenario in which no patents are awarded for those subclasses.



## Graphical diagnostics and tests

### ▶ Example 4: DID diagnostic graphs and tests

As illustrated in [example 1](#), when conducting a DID study, it is common to complement the regression analysis with graphical diagnostics and tests that provide evidence of whether an estimated effect can be given a causal interpretation. As discussed in [\[TE\] DID intro](#), we would like to observe that the treated and control groups had mean outcomes that evolved similarly to each other over time prior to the treatment. This is usually referred to as a parallel-trends or common-trends assumption. We would also like to ascertain that neither the control nor the treatment group changed their behavior in anticipation of the treatment. This is assessed using a Granger-type test.

Below, we use simulated data to illustrate the diagnostics and tests available after `didregress` and `xtdidregress`. Simulated data helps us know exactly what we should expect and how to interpret it.

Suppose we have a panel dataset with 10 time points, denoted by  $t_1$ , where a treatment has taken place between  $t_1 = 5$  and  $t_1 = 6$ . We have a set of covariates,  $x_1$  and  $x_2$ , and an outcome,  $y_1$ . The outcome could be something like patient satisfaction, as in [example 1](#), or the number of patents filed by U.S. inventors, as in [example 3](#).



We fit the model:

```
. use https://www.stata-press.com/data/r17/parallelt
(Simulated data to test parallel-trends assumption)
. xtset id1
Panel variable: id1 (unbalanced)
. xtddidregress (y1 c.x1##c.x2) (treated1), group(id1) time(t1)
Number of groups and treatment time
Time variable: t1
Control:      treated1 = 0
Treatment:    treated1 = 1
```

		Control	Treatment
Group	id1	102	98
Time	Minimum	1	6
	Maximum	1	6

Difference-in-differences regression Number of obs = 2,000  
 Data type: Longitudinal

(Std. err. adjusted for 200 clusters in id1)

y1		Coefficient	Robust std. err.	t	P> t	[95% conf. interval]
ATET	treated1 (Treated vs Untreated)	.5069426	.0220218	23.02	0.000	.4635166 .5503686

Note: ATET estimate adjusted for covariates, panel effects, and time effects.

Is this result valid? We can first explore the assumption of parallel trends graphically, comparing the trajectories of the outcome variable for the control and treatment groups prior to the date of treatment. We can check this assumption by plotting the means of the outcome over time for both groups or by visualizing the results of the linear-trends model. We can perform both of these diagnostic checks by using `estat trendplots`.

```
. estat trendplots
```

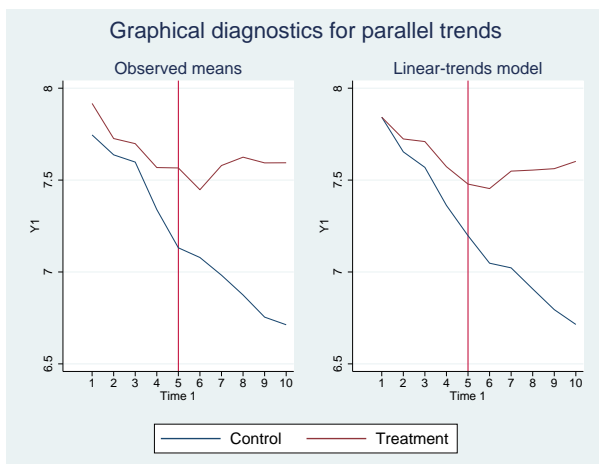


Figure 2

Looking at the plotted observed means (left side of figure 2), the outcome trajectories in the control and treatment groups prior to the treatment are somewhat different. While we can observe declining trends in both groups, it looks as though the decline is progressing more rapidly in the control group, especially between time points three and five. We can get a clearer picture of this by looking at the results of the linear-trends model on the right side of figure 2. The group-level trajectories are shown with respect to a common reference point,  $\tau_1 = 1$ , which makes it easy to discern whether they are parallel. In this case, we can see that they are not. The differences between the treatment and control groups are growing larger over time up to  $\tau_1 = 5$  (posttreatment time periods are not relevant when assessing the parallel-trends assumption). Judging by figure 2 alone, we should be concerned about whether the parallel-trends assumption holds for our effect estimate.

A more formal way to assess whether the pretreatment trajectories are parallel is to perform a test on the linear-trends model coefficient that captures the differences in the trends between treated and controls. If the pretreatment trends are actually linear in both groups, then this coefficient will be 0 because there are no differences in the slopes between the two groups. Thus, by testing this coefficient against 0, we have a test of the null hypothesis that the pretreatment period trajectories are parallel. We can perform this test by using `estat ptrends`:

```
. estat ptrends
Parallel-trends test (pretreatment time period)
H0: Linear trends are parallel
F(1, 199) = 39.97
Prob > F = 0.0000
```

We reject the null hypothesis of the linear trends being parallel.

We can also think of nonparallel as an indication of an anticipatory treatment effect. We saw that the trends were not parallel before the treatment took place, which could indicate a treatment effect even before the treatment is implemented. Thus, another way to state our parallel-trends assumption is that there should be no treatment effect in anticipation of the treatment. To test this assumption, we could fit a Granger-type causality model where we augment our model with dummies that indicate future treatment status for each time period prior to the treatment. A joint test of the coefficients on these dummies against 0 can be used as a test of the null hypothesis that no anticipatory effects have taken place. We can perform this test by using `estat granger`:

```

. estat granger
Granger causality test
H0: No effect in anticipation of treatment
F(4, 199) = 18.17
Prob > F = 0.0000

```

The result suggests that we reject the null hypothesis of no anticipatory effects prior to treatment. Thus, based on the results of both `estat ptrends` and `estat granger`, we conclude that we should be concerned about identification of ATET.

Notice that the parallel-trends  $F$  test consumes only 1 numerator degree of freedom, while the Granger causality  $F$  test consumes 4. That is because we have five pretreatment time periods and thus four coefficients to test. In this example, both tests would be appropriate, but the parallel-trends test has higher statistical power. However, as we will see below, the less-powered Granger test is more flexible and can be used in situations where the differences between treatment and control groups are nonlinear in pretreatment periods.

◀

### ► Example 5: DID diagnostics and tests with nonlinear mean outcome differences

We now look at a different outcome and policy using the same simulated dataset as in [example 4](#). We fit the same model but with the new variables:

```

. xtset id2
Panel variable: id2 (balanced)
. xtdidregress (y2 c.z1##c.z2) (treated2), group(id2) time(t2)
Number of groups and treatment time
Time variable: t2
Control:      treated2 = 0
Treatment:    treated2 = 1

```

		Control	Treatment
Group	id2	480	520
Time	Minimum	1	6
	Maximum	1	6

Difference-in-differences regression Number of obs = 10,000  
 Data type: Longitudinal

(Std. err. adjusted for 1,000 clusters in id2)

y2	Robust			t	P> t	[95% conf. interval]
	Coefficient	std. err.				
ATET treated2 (Treated vs Untreated)	.2636651	.0097188	27.13	0.000	.2445936	.2827367

Note: ATET estimate adjusted for covariates, panel effects, and time effects.

Here is the diagnostic plot produced by `estat trendplots`:

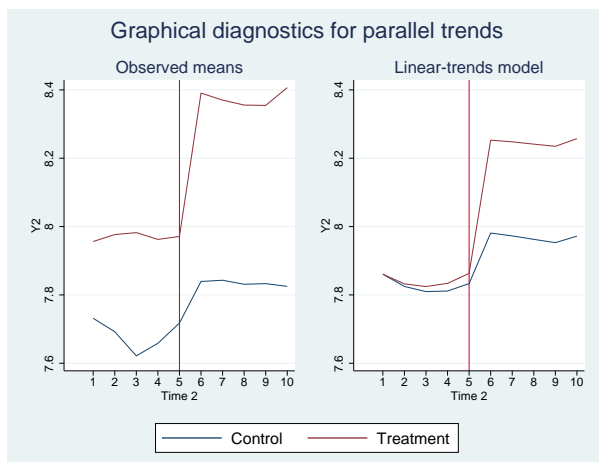


Figure 3

Inspecting the observed means over the pretreatment time periods, we can see that no considerable change has occurred in the outcome of the treatment group over time. For the control group, on the other hand, we observe a somewhat U-shaped trajectory in advance of the treatment. However, if we look at the plot from the linear-trends model, both trajectories appear somewhat U-shaped, and the trends appear to be almost parallel. Indeed, if we use `estat ptrends`, we fail to reject the null hypothesis of parallel trends:

```
. estat ptrends
Parallel-trends test (pretreatment time period)
H0: Linear trends are parallel
F(1, 999) = 2.13
Prob > F = 0.1446
```

The Granger causality test can handle cases in which the trajectories are nonlinear and for which `estat ptrends` will fail to reject the null hypothesis when it should:

```
. estat granger
Granger causality test
H0: No effect in anticipation of treatment
F(4, 999) = 9.86
Prob > F = 0.0000
```

We correctly reject the null hypothesis of no effect in anticipation of the treatment.

► Example 6: Diagnostics and tests when parallel-trends assumption is satisfied

Finally, we look at a case where the assumption of parallel trends is satisfied. Again, we use the simulated dataset from [example 4](#). We fit the following model:

```
. xtset id3
Panel variable: id3 (balanced)
. xtddidregress (y3 c.w1##c.w2) (treated3), group(id3) time(t3)
Number of groups and treatment time
Time variable: t3
Control:      treated3 = 0
Treatment:    treated3 = 1
```

		Control	Treatment
Group	id3	502	498
Time	Minimum	1	6
	Maximum	1	6

Difference-in-differences regression Number of obs = 10,000  
 Data type: Longitudinal  
(Std. err. adjusted for 1,000 clusters in id3)

y3		Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
ATET	treated3 (Treated vs Untreated)	.4996049	.0102458	48.76	0.000	.4794991	.5197107

Note: ATET estimate adjusted for covariates, panel effects, and time effects.

Using estat trendplots, we obtain

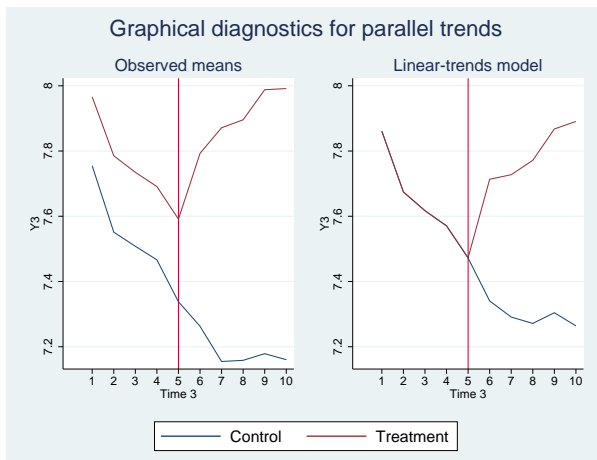


Figure 4

This time, the trajectories of the observed means appear to be parallel before the treatment occurs. Both follow a declining trend up to the last pretreatment time point. Looking at the results from the linear-trends model, the pretreatment trajectories appear to be the same. Using `estat ptrends` yields the following result:

```
. estat ptrends
Parallel-trends test (pretreatment time period)
H0: Linear trends are parallel
F(1, 999) = 0.00
Prob > F = 0.9688
```

We cannot reject the null hypothesis of parallel linear trends. Likewise, using `estat granger`, we do not reject the null hypothesis of the absence of anticipatory effects:

```
. estat granger
Granger causality test
H0: No effect in anticipation of treatment
F(4, 999) = 0.52
Prob > F = 0.7220
```

◀

## ▷ Example 7: Time-specific treatment effects

As we have seen in the previous examples, we can use `estat granger` to test for treatment effects occurring prior to the point at which treatment has been administered. Beyond the particular purpose of this test, however, it can be instructive to inspect not only pretreatment effects but also posttreatment effects that are allowed to vary over time. That is, rather than relying on a single treatment-effect estimate that is assumed to be constant over time, it can be instructive to check whether treatment effects change as time unfolds. We can fit a model that includes lags and leads of an indicator for the time period in which the treatment began. The coefficients on the lags can be used to assess whether there is any change in the treatment effect during posttreatment time periods. We can use the postestimation command `estat grangerplot` to fit such a model and plot the resulting coefficients. This model is also sometimes referred to as an event study model in the literature; see [Schmidheiny and Siegloch \(2019\)](#) and [Clarke and Tapia-Schyte \(2021\)](#).

We start by fitting the following DID model:

```
. use https://www.stata-press.com/data/r17/hospdd
(Artificial hospital admission procedure data)
. didregress (satis) (procedure), group(hospital) time(month)
Number of groups and treatment time
Time variable: month
Control:      procedure = 0
Treatment:    procedure = 1
```

	Control	Treatment
Group hospital	28	18
Time		
Minimum	1	4
Maximum	1	4

Difference-in-differences regression Number of obs = 7,368  
 Data type: Repeated cross-sectional  
 (Std. err. adjusted for 46 clusters in hospital)

satis	Robust		t	P> t	[95% conf. interval]	
	Coefficient	std. err.				
ATET procedure (New vs Old)	.8479879	.0321121	26.41	0.000	.7833108	.912665

Note: ATET estimate adjusted for group effects and time effects.

We can now simply use `estat grangerplot` to plot pre- and postintervention treatment effects:

```
. estat grangerplot
```

This yields the following graph:

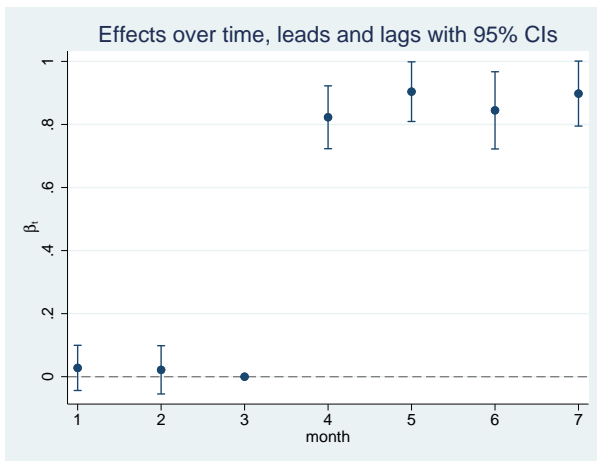


Figure 5

By default, `estat grangerplot` uses all available leads and lags, and in this case, we have estimates for two leads and four lags. For identification purposes, we must omit one of the indicator variables. By default, `estat grangerplot` omits the first lead, which sets the coefficient to 0 that corresponds to the time period prior to treatment administration. Thus, the plotted effects are normalized with respect to that time period. If we wanted, we could change the baseline period by using the `baseline()` option. We can see that the coefficients on the leads that correspond to the first two months are both close to 0, which shows that there are no substantial effects in the pretreatment era. The posttreatment effects for months 4 to 7 range between around 0.8 and 0.9. When we account for the uncertainty of these estimates, as indicated by the plotted 95% confidence intervals, it appears as though the treatment effects are rather stable over time in this example.

Rather than just plotting the coefficients, we could be interested in looking at a table with the numeric results. We can obtain the underlying results by specifying the `verbose` option. We are also using the `nodraw` option to see only the numeric results and not draw the graph again:

```
. estat grangerplot, nodraw verbose
Linear regression, absorbing indicators          Number of obs   =   7,368
Absorbed variable: hospital                   No. of categories =    46
                                                F(12, 45)      =   94.68
                                                Prob > F       =   0.0000
                                                R-squared     =   0.5334
                                                Adj R-squared =   0.5298
                                                Root MSE     =   0.7240

                                (Std. err. adjusted for 46 clusters in hospital)
```

satis	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
month						
February	-.007044	.0263953	-0.27	0.791	-.0602068	.0461188
March	.0335696	.0255925	1.31	0.196	-.0179764	.0851156
April	.0187852	.0250623	0.75	0.457	-.0316927	.0692632
May	-.0211152	.0269569	-0.78	0.438	-.0754092	.0331788
June	.0091208	.0179016	0.51	0.613	-.026935	.0451766
July	-.0203444	.0306266	-0.66	0.510	-.0820296	.0413407
_lead3	.027897	.035569	0.78	0.437	-.0437426	.0995367
_lead2	.0217322	.0380076	0.57	0.570	-.054819	.0982833
_lag0	.8228153	.0494933	16.62	0.000	.7231307	.9224999
_lag1	.9040498	.0469682	19.25	0.000	.8094511	.9986486
_lag2	.844724	.0608006	13.89	0.000	.7222654	.9671826
_lag3	.8978885	.0511588	17.55	0.000	.7948494	1.000928
_cons	3.433074	.0198449	173.00	0.000	3.393104	3.473044

The coefficients on the variables that begin with `_lead` and `_lag` are the ones that are plotted in our graph.

As an interesting side note, notice that our original DID model is a special case of the model that we fit with `estat grangerplot`. Specifically, it is the model that includes only a single lead and no lags. To reproduce our original DID result, we can use the `nleads()` and `nlags()` options:



```
. estat grangerplot, nodraw verbose nleads(1) nlags(0)
Linear regression, absorbing indicators      Number of obs      = 7,368
Absorbed variable: hospital                No. of categories  = 46
                                           F(7, 45)           = 138.73
                                           Prob > F           = 0.0000
                                           R-squared          = 0.5333
                                           Adj R-squared      = 0.5299
                                           Root MSE          = 0.7238

                                           (Std. err. adjusted for 46 clusters in hospital)
```

satis	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
month						
February	-.0096077	.0184317	-0.52	0.605	-.0467311	.0275158
March	.0219686	.018251	1.20	0.235	-.0147907	.0587279
April	-.0032839	.0221028	-0.15	0.883	-.0478013	.0412335
May	-.0094027	.0232399	-0.40	0.688	-.0562103	.0374048
June	-.0038375	.0190634	-0.20	0.841	-.0422332	.0345581
July	-.0111941	.0230029	-0.49	0.629	-.0575244	.0351361
_lag0	.8479879	.0321121	26.41	0.000	.7833108	.912665
_cons	3.444675	.011354	303.39	0.000	3.421807	3.467543

As we can see, the results match the ones we obtained earlier with didregress.

Notice also that the model used by estat granger is a special case as well; it is the model with all available leads and no lags. To re-create the test result from estat granger, we use the post option to store the results:

```
. estat grangerplot, nodraw verbose post nlags(0)
Linear regression, absorbing indicators      Number of obs      = 7,368
Absorbed variable: hospital                No. of categories  = 46
                                           F(9, 45)           = 113.78
                                           Prob > F           = 0.0000
                                           R-squared          = 0.5333
                                           Adj R-squared      = 0.5298
                                           Root MSE          = 0.7239

                                           (Std. err. adjusted for 46 clusters in hospital)
```

satis	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
month						
February	-.007044	.0263899	-0.27	0.791	-.0601959	.0461079
March	.0335696	.0255873	1.31	0.196	-.0179658	.085105
April	.0002573	.0243547	0.01	0.992	-.0487955	.0493101
May	-.0058616	.0249982	-0.23	0.816	-.0562105	.0444874
June	-.0002964	.0199414	-0.01	0.988	-.0404605	.0398678
July	-.007653	.0251828	-0.30	0.763	-.0583738	.0430679
_lead3	.027897	.0355617	0.78	0.437	-.043728	.099522
_lead2	.0217322	.0379998	0.57	0.570	-.0548033	.0982676
_lag0	.8673694	.0424929	20.41	0.000	.7817844	.9529544
_cons	3.433074	.0198408	173.03	0.000	3.393113	3.473035

We can now test whether the coefficients on the leads are jointly 0:

```
. test _lead3 _lead2
( 1)  _lead3 = 0
( 2)  _lead2 = 0
      F( 2, 45) = 0.33
      Prob > F = 0.7239
```

The test result indicates that we cannot reject the null hypothesis that the coefficients are jointly 0. We fit our original model again to compare the results with the ones from `estat granger`:

```
. quietly didregress (satis) (procedure), group(hospital) time(month)
. estat granger
Granger causality test
H0: No effect in anticipation of treatment
F(2, 45) = 0.33
Prob > F = 0.7239
```

As we can see, the results are the same. Using `estat grangerplot` followed by `test` allows us to perform additional tests of lags and leads that are not available through `estat granger`.

◀

## Specifying a 2-by-2 DID

### ► Example 8: Specifying a 2-by-2 DID

`didregress` and `xtdidregress` by default fit generalized DID models, also known as two-way fixed-effects models. Yet DID models sometimes are viewed from a two-period and two-group perspective, a 2-by-2 DID. You can also fit a 2-by-2 DID using `didregress` and `xtdidregress`. In fact, you will get equivalent results using a 2-by-2 DID or a generalized DID. The generalized DID is the default because it allows for a wider range of specifications that would not be feasible within the 2-by-2 framework, such as cases when the intervention occurs at different points in time.

In [example 1](#), we got the ATET for hospitals that instituted a new admissions procedure. We typed the following:

```
. use https://www.stata-press.com/data/r17/hospdd
. didregress (satis) (procedure), group(hospital) time(month)
```

This implies that we are regressing `satis` on `procedure` and indicators for hospitals and for months. The indicators are created and added as regressors to our model by default. To fit a 2-by-2 model, we need to omit the hospital and month indicators and instead add an indicator for the period after treatment and an indicator for groups that are treated.

This is what we do below. We create the treatment-group indicator based on the fact that the variable `procedure` is 1 for individuals that experienced the new procedure and 0 otherwise; this variable identifies treated hospitals. We create the posttreatment indicator based on the fact that the new procedures are established after March, which has a value of 3 in the data.

```
. bysort hospital: egen treated = max(procedure)
. generate post = month>3
```

We now fit

```
. didregress (satis i.treated i.post) (procedure), nogteffects
> group(hospital) time(month)
Number of groups and treatment time
Time variable: month
Control:      procedure = 0
Treatment:    procedure = 1
```

	Control	Treatment
Group hospital	28	18
Time Minimum	1	4
Maximum	1	4

```
Difference-in-differences regression          Number of obs = 7,368
Data type: Repeated cross-sectional
              (Std. err. adjusted for 46 clusters in hospital)
```

satis	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
ATET procedure (New vs Old)	.8479879	.0320051	26.50	0.000	.7835263	.9124494

Note: ATET estimate adjusted for covariates.

We add the new indicators to our list of covariates and use the `nogteffects` option to exclude the group and time indicators that were included by default.

The point estimates are identical. What changes are the standard errors. They change because we have a different number of regressors.

◀

## Standard-error considerations

### ▷ Example 9: Standard errors and data aggregation

In [example 3](#), we had 336 treated subclasses and 7,248 subclasses in total. However, many studies face the challenge of having very few elements per group. In fact, sometimes the data analyzed consist of only two groups—the treatment group in which treatment is administered to members of the group and the controls group in which no treatment is administered. Think, for example, of an analysis at the state level where one state is treated and one state is controlled. As discussed in [\[TE\] DID intro](#), these scenarios with few elements pose a challenge for inference. For a good discussion on these issues, see [MacKinnon \(2019\)](#).

`didregress` and `xtdidregress` provide alternatives in such cases. One alternative is to compute standard errors by using the wild cluster bootstrap. Another alternative is to use bias-corrected clustered standard errors with the degrees-of-freedom adjustment proposed by [Bell and McCaffrey \(2002\)](#). A final alternative is to aggregate your data and then compute effects and standard errors such as proposed by [Donald and Lang \(2007\)](#) or use bias-corrected standard errors.

Below, we explore these options using simulated data. We created a dataset with 2,000 individuals and five time periods. The treatment occurs at the county level and there are six counties. Two of the counties, county 1 and county 2, receive the treatment and the remaining four counties do not. The true value of the ATET is  $-1.0$ . As we see below, the groups are unbalanced.

```
. use https://www.stata-press.com/data/r17/smallg
(Simulated data with a small number of groups)
```

```
. tab county
```

County	Freq.	Percent	Cum.
1	715	7.15	7.15
2	2,570	25.70	32.85
3	3,410	34.10	66.95
4	2,285	22.85	89.80
5	920	9.20	99.00
6	100	1.00	100.00
Total	10,000	100.00	

First, we fit the model using the default standard errors, which perform well with many balanced groups but not necessarily well with data like these. We compare the other results with these results.

```
. didregress (outcome x i.b) (treated), group(county) time(year)
```

```
Number of groups and treatment time
```

```
Time variable: year
```

```
Control:      treated = 0
```

```
Treatment:   treated = 1
```

	Control	Treatment
Group		
county	4	2
Time		
Minimum	2011	2013
Maximum	2011	2013

```
Difference-in-differences regression
```

```
Number of obs = 10,000
```

```
Data type: Repeated cross-sectional
```

```
(Std. err. adjusted for 6 clusters in county)
```

outcome	Robust		t	P> t	[95% conf. interval]	
	Coefficient	std. err.				
ATET						
treated						
(Treated						
vs						
Untreated)	-.9394987	.0884134	-10.63	0.000	-1.166773	-.7122247

Note: ATET estimate adjusted for covariates, group effects, and time effects.

For this draw, the point estimates are reasonably close to the true value of  $-1.0$ , and the true values are contained inside the confidence interval.

Next, we use the wild cluster bootstrap. For a good introduction to the methodology, see [Cameron, Gelbach, and Miller \(2008\)](#), [MacKinnon and Webb \(2018\)](#), and [Roodman et al. \(2019\)](#). The wild cluster bootstrap works well in scenarios like the one above, where there are few groups.

We use the wild cluster bootstrap to construct  $p$ -values and confidence intervals, imposing the null hypothesis that the ATET is 0. We describe both computations in *Methods and formulas*. Here are the results:

```
. didregress (outcome x i.b) (treated), group(county) time(year)
> wildbootstrap(rseed(123) errorweight(webb))
computing 1000 replications

Confidence interval lower bound
.....
Confidence interval upper bound
.....

Number of groups and treatment time
Time variable: year
Control:      treated = 0
Treatment:    treated = 1
```

		Control	Treatment
Group	county	4	2
Time	Minimum	2011	2013
	Maximum	2011	2013

DID with wild-cluster bootstrap inference

Number of obs = 10,000  
 No. of clusters = 6  
 Replications = 1,000

Data type: Repeated cross-sectional  
 Error weight: webb

outcome	Coefficient	t	P> t	[95.10% conf. interval]	
ATET treated (Treated vs Untreated)	-.9394987	-10.63	0.020	-1.248532	-.5621484

Note: 95.10% confidence interval is wider than requested.

Note: ATET estimate adjusted for covariates, group effects, and time effects.

Above, we first see the iterations used to find the confidence interval lower bound and upper bound. The optimization algorithm sometimes converges to a level that is below the test size. In such cases, the confidence interval is conservative. In the example above, instead of a 95% confidence interval, you obtain a 95.1% confidence interval. If there are  $t$  statistics that have exactly the same value across bootstrap replications, the algorithm will not solve exactly for the requested confidence level. We also use `errorweight()` with error weight `webb` to compute the wild bootstrap. This is best in cases with less than 10 groups, as suggested by [Roodman et al. \(2019\)](#). The results from the wild bootstrap suggest more uncertainty than the default confidence interval.

Now we look at two data aggregation methods. In both cases, data aggregation occurs in two steps. In the first step, we regress the outcome on the set of covariates that vary at the individual, group, and time levels. We then estimate the group–time fixed effects from this procedure to use in a second stage as the new dependent variable. In the second stage, we aggregate the remaining covariates and the new dependent variable at the group and time levels and run a regression. This aggregation strategy is described in more detail in [Bertrand, Duflo, and Mullainathan \(2004\)](#), [Donald and Lang \(2007\)](#), and [Cameron and Miller \(2015\)](#). We also describe the aggregation methods in *Methods and formulas*.

Below, we aggregate the data as described above and estimate bias-corrected standard errors by using the degrees-of-freedom adjustment suggested by [Bell and McCaffrey \(2002\)](#). This method of getting standard errors is computationally intensive, so it is well suited for cases where the dimensionality of the problem is reduced via aggregation.

```
. didregress (outcome x i.b) (treated), group(county) time(year)
> aggregate(standard) vce(hc2)
Computing degrees-of-freedom:
treated .
```

Number of groups and treatment time

Time variable: year  
 Control: treated = 0  
 Treatment: treated = 1

	Control	Treatment
Group		
county	4	2
Time		
Minimum	2011	2013
Maximum	2011	2013

Difference-in-differences regression

Number of obs = 30  
 No. of clusters = 6

Data type: Repeated cross-sectional  
 Aggregation: Standard

outcome	Robust HC2		t	P> t	[95% conf. interval]	
	Coefficient	std. err.				
ATET						
treated						
(Treated						
vs						
Untreated)	-.9958521	.1373277	-7.25	0.017	-1.566242	-.4254624

Note: ATET estimate adjusted for covariates, group effects, and time effects.

We see that the confidence intervals are again wider than with the default standard errors. It is also worth noticing that we went from 10,000 observations to 30. This is because we aggregated at the county and year levels. We have six counties and five years.

We could also aggregate our data and compute the  $t$  statistics using the degrees of freedom proposed by Donald and Lang (2007). This gives us

```
. didregress (outcome x i.b) (treated), group(county) time(year)
> aggregate(dlang)

Number of groups and treatment time
Time variable: year
Control:      treated = 0
Treatment:    treated = 1
```

		Control	Treatment
Group	county	4	2
Time	Minimum	2011	2013
	Maximum	2011	2013

```
Difference-in-differences regression          Number of obs = 30
Data type:  Repeated cross-sectional
Aggregation: Donald-Lang
```

outcome	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
ATET treated (Treated vs Untreated)	-.9958521	.1224496	-8.13	0.000	-1.248576	-.7431287

Note: ATET estimate adjusted for covariates, group effects, and time effects.

Again, we see wider confidence intervals.

Whenever you have few elements for each group and the groups are unbalanced, as in this example, you should be careful not to base your conclusions solely on the default cluster-robust standard errors. As was mentioned in [TE] **DID intro**, you should validate your conclusions by using one of the standard error computations suggested above.

**Default cluster-robust standard errors**▷ **Example 10: Default cluster-robust standard errors with `didregress` and `xtdidregress`**

We have stated that `didregress` was designed to handle repeated cross-sections and `xtdidregress` was designed to handle panel datasets. However, you can use `xtdidregress` when you have repeated cross-sections. For instance, for the hospital dataset in [example 1](#), you may have typed

```
. use https://www.stata-press.com/data/r17/hospdd
(Artificial hospital admission procedure data)
. xtset hospital
Panel variable: hospital (unbalanced)
. xtdidregress (satis) (procedure), group(hospital) time(month)
Number of groups and treatment time
Time variable: month
Control:      procedure = 0
Treatment:    procedure = 1
```

	Control	Treatment
Group		
hospital	28	18
Time		
Minimum	1	4
Maximum	1	4

```
Difference-in-differences regression
Data type: Longitudinal
```

Number of obs = 7,368

(Std. err. adjusted for 46 clusters in hospital)

	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
ATET						
procedure						
(New						
vs						
Old)	.8479879	.0320138	26.49	0.000	.7835088	.9124669

Note: ATET estimate adjusted for panel effects and time effects.

You will get the same point estimate as you would get with `didregress` but different standard errors. The reason is that `xtdidregress` with cluster-robust standard errors does not count the group variables as regressors in the degrees-of-freedom correction used to compute standard errors. It relies on the asymptotic theory of fixed-effects regression, where the number of group effects are expected to grow with the sample size. Put differently, `xtdidregress` is using `xtreg, fe` to compute the default cluster-robust standard errors. `didregress` uses `areg`, which assumes the number of groups is fixed and counts them in the degrees-of-freedom computation.

As a practical matter, these standard errors are going to be close to each other when the number of observations per cluster is large, as is the case for this example. As the number of observations per cluster becomes smaller, however, the standard errors will differ more, with standard errors of `didregress` tending to be larger.

Which estimator to use depends on the assumptions you would like to make about your data. If you think your data should be treated as panel data, with the number of groups growing with the sample size, then even if you have repeated cross-sections, you should use `xtdidregress`.



More in-depth discussions about cluster-robust standard errors can be found in [Cameron and Miller \(2015\)](#) and [Wooldridge \(2010\)](#).

◀

## Stored results

`didregress` stores the following in `e()`:

### Scalars

<code>e(N)</code>	number of observations
<code>e(N_clust)</code>	number of clusters
<code>e(tmin)</code>	minimum of first observed treatment time across groups
<code>e(tmax)</code>	maximum of first observed treatment time across groups
<code>e(N_reps)</code>	number of bootstrap replications
<code>e(df_r)</code>	residual degrees of freedom
<code>e(blocksize)</code>	block size used in wild bootstrap computations

### Macros

<code>e(cmd)</code>	<code>didregress</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(treatment)</code>	indicator for treated observations
<code>e(treatment_type)</code>	binary or continuous
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(wb_weight)</code>	wild bootstrap error weight
<code>e(datatype)</code>	data type
<code>e(groupvars)</code>	group variables
<code>e(clustvar)</code>	name of cluster variable
<code>e(timevar)</code>	time variable
<code>e(vce)</code>	<i>vctype</i> specified in <code>vce()</code>
<code>e(vctype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(_contrast_not_ok)</code>	prediction disallowed by <code>contrast</code>

### Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance-covariance matrix of the estimators
<code>e(group_count)</code>	matrix with number of groups and treatment time
<code>e(fwboot)</code>	wild bootstrap estimates
<code>e(wboot)</code>	wild bootstrap statistics and constraint
<code>e(aggmethod)</code>	aggregation method

### Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

### Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -values, and confidence intervals
-----------------------	--

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r`-class command is run after the estimation command.

`xtdidregress` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(N_clust)</code>	number of clusters
<code>e(tmin)</code>	minimum of first observed treatment time across groups
<code>e(tmax)</code>	maximum of first observed treatment time across groups
<code>e(N_reps)</code>	number of bootstrap replications
<code>e(df_r)</code>	residual degrees of freedom
<code>e(blocksize)</code>	block size used in wild bootstrap computations

Macros

<code>e(cmd)</code>	<code>xtdidregress</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(treatment)</code>	indicator for treated observations
<code>e(treatment_type)</code>	binary or continuous
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(wb_weight)</code>	wild bootstrap error weight
<code>e(datatype)</code>	data type
<code>e(groupvars)</code>	group variables
<code>e(clustvar)</code>	name of cluster variable
<code>e(panelvar)</code>	panel variable
<code>e(timevar)</code>	time variable
<code>e(vce)</code>	<i>vctype</i> specified in <code>vce()</code>
<code>e(vctype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(_contrast_not_ok)</code>	prediction disallowed by <code>contrast</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance-covariance matrix of the estimators
<code>e(group_count)</code>	matrix with number of groups and treatment time
<code>e(fwboot)</code>	wild bootstrap estimates
<code>e(wboot)</code>	wild bootstrap statistics and constraint
<code>e(aggmethod)</code>	aggregation method

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -values, and confidence intervals
-----------------------	--

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

## Methods and formulas

Methods and formulas are presented under the following headings:

*DID with repeated cross-sectional data*

*DDD model*

*DID and DDD models with longitudinal data*

*Aggregation estimators*

*Wild bootstrap confidence intervals and p-values*

*Bias-corrected clustered standard error*

## DID with repeated cross-sectional data

The DID model for repeated cross-sectional data fit by `didregress` is given by

$$y_{ist} = \gamma_s + \gamma_t + \mathbf{z}_{ist}\beta + D_{st}\delta + \varepsilon_{ist} \quad (1)$$

where  $i$  is the observation-level index,  $s$  is a group-level index, and  $t$  is a time-level index. For example, we might have yearly repeated cross-sectional data for individuals living in different states; in this case,  $i$  denotes the individual,  $s$  the state, and  $t$  the year. In (1) above,  $\gamma_s$  are group fixed effects and  $\gamma_t$  are time fixed effects.  $\mathbf{z}_{ist}$  are the covariates, and  $\varepsilon_{ist}$  is the error term.  $D_{st}$  is the treatment that varies at the group and time levels.  $D_{st}$  could be binary or continuous.

If  $y_{ist}$  is `y` in the data, then  $\mathbf{z}_{ist}$  are `z1` and `z2`,  $D_{st}$  is `d`, the group is `state`, and time is `year`. To fit the model, you would type

```
didregress (y z1 z2) (d), group(state) time(year)
```

which is equivalent to typing

```
areg y z1 z2 i.year d, absorb(state) vce(cluster state)
```

The methods and formulas for this model are those of a linear regression and can be found in [Methods and formulas](#) in [R] `areg`.

## DDD model

The DDD model is given by

$$y_{isgt} = \gamma_s + \gamma_g + \gamma_t + \gamma_s\gamma_t + \gamma_g\gamma_t + \gamma_s\gamma_g + \mathbf{z}_{isgt}\beta + D_{sgt}\delta + \varepsilon_{isgt} \quad (2)$$

where  $i$  is the observation-level index,  $s$  and  $g$  are group-level indices, and  $t$  is a time-level index. For example, we might have yearly repeated cross-sectional data for older and younger individuals living in different states. In this case,  $i$  denotes the individual,  $s$  the state,  $g$  age group, and  $t$  the year. In (2) above,  $\gamma_s$  are group  $s$  fixed effects,  $\gamma_g$  are group  $g$  fixed effects, and  $\gamma_t$  are time fixed effects.  $\mathbf{z}_{isgt}$  are the covariates, and  $\varepsilon_{isgt}$  is the error term.  $D_{sgt}$  denotes the treatment that varies at the group  $s$ , group  $g$ , and time levels.  $D_{sgt}$  could be binary or continuous.

Say you now want to fit a DDD, where the treatment occurs for `group2` in some of the states, `state`. You would type

```
didregress (y z1 z2) (d), group(state group2) time(year)
```

`didregress` would construct the group and time interactions in (2) and fit a linear regression. The methods and formulas for the specification are the same as those in [Methods and formulas](#) in [R] `areg`.

## DID and DDD models with longitudinal data

The DID model for longitudinal data fit by `xtdidregress` is given by

$$y_{ist} = \alpha_i + \gamma_t + \mathbf{z}_{ist}\beta + D_{st}\delta + \varepsilon_{ist} \quad (3)$$

where  $i$  is the observation-level index,  $s$  is a group-level index, and  $t$  is a time-level index. For example, we might have a longitudinal dataset of individuals over time living in a given state; in this case,  $i$  denotes the individual,  $s$  the state, and  $t$  the year. In (3) above,  $\alpha_i$  are individual fixed effects and  $\gamma_t$  are time fixed effects.  $\mathbf{z}_{ist}$  are the covariates, and  $\varepsilon_{ist}$  is the error term.  $D_{st}$  denotes the treatment that varies at the group and time levels.  $D_{st}$  could be binary or continuous. Individuals,  $i$ , are assumed to be nested within the group; thus, group effects are subsumed by the individual effects.

If  $y_{ist}$  is  $y$  in the data, then  $\mathbf{z}_{ist}$  are  $\mathbf{z}_1$  and  $\mathbf{z}_2$ ,  $D_{st}$  is  $d$ , the group is `state`, time is `year`, and individuals are denoted by `id`. To fit the model, you would type

```
xtset id year
xtdidregress (y z1 z2) (d), group(state) time(year)
```

which is equivalent to typing

```
xtreg y z1 z2 i.year d, fe vce(cluster state)
```

The methods and formulas for this model are those of a within estimator computed by `xtreg`, `fe` and can be found in [Methods and formulas](#) in [XT] `xtreg`.

The DDD model is given by

$$y_{isgt} = \alpha_i + \gamma_t + \gamma_t \gamma_s + \gamma_t \gamma_g + \mathbf{z}_{ist} \beta + D_{st} \delta + \varepsilon_{isgt}$$

## Aggregation estimators

To discuss the aggregation estimators, it is instructive to rewrite the covariate vector  $\mathbf{z}_{ist}$  as consisting of group and time invariant components  $\mathbf{z}_{1st}$  and time-, group-, and individual-varying components  $\mathbf{z}_{2ist}$ . The DID model can now be expressed as

$$y_{ist} = \gamma_s + \gamma_t + \mathbf{z}_{1st} \beta_1 + \mathbf{z}_{2ist} \beta_2 + D_{st} \delta + \varepsilon_{ist} \quad (4)$$

$$C_{st} = \gamma_s + \gamma_t + \mathbf{z}_{1st} \beta_1 + D_{st} \delta + \nu_{st} \quad (5)$$

The `standard` and `dlang`, `constant` aggregation methods regress  $y_{ist}$  on  $\mathbf{z}_{2ist}$  in (4) and then obtain an estimate of  $C_{st}$ ,  $\hat{C}_{st}$ . After we regress  $y_{ist}$  on  $\mathbf{z}_{2ist}$ , we get an estimate of the group–time effects  $\hat{C}_{st}$  from  $y_{ist} - \mathbf{z}_{2ist} \beta_2 - \hat{\varepsilon}_{ist}$ . These effect estimates come from using `predict` with option `d` after `areg`. We then aggregate the data at the  $s$  and  $t$  levels. With the aggregated data, we run a regression of  $\hat{C}_{st}$  on  $\mathbf{z}_{1st}$  and on  $D_{st}$  using group and time fixed effects in (5). The `dlang` method computes the ordinary least-squares standard errors from this second stage. The `standard` method uses cluster–robust standard errors clustered at the group level by default.

The `dlang`, `varying` method runs a regression for each group defined by  $s$  and  $t$  and obtains an estimate of  $C_{st}$  as the set of constants for each regression. Whereas in the first case  $\beta_2$  was the same vector across the sample, in this case, we will get a different estimate of the slope coefficient for each group. The second step is the same as for `dlang`, `constant`.

## Wild bootstrap confidence intervals and p-values

The wild cluster bootstrap is a bit different from the conventional bootstrap in that it keeps the covariates fixed and constructs the new dependent variables using the residuals from the original regression. The wild bootstrap procedure proceeds as follows:

1. Fit a restricted model, where the ATET is constrained to be 0. In other words, imposing the null hypothesis that the ATET is 0. Obtain the predicted values,  $\hat{y}_r$ , and residuals,  $\hat{u}_r$ , from the restricted model, where the subscript  $r$  refers to the bootstrap replicates. Also fit the unrestricted model and compute the  $t$  statistic,  $t$ , testing the null hypothesis ATET = 0.

2. At each of the subsequent  $B - 1$  bootstrap steps, compute  $\mathbf{y}^* = \widehat{\mathbf{y}}_r + \widehat{\mathbf{u}}_r \odot (\mathbf{K}\mathbf{w})$ , where  $\mathbf{w}$  is a column vector of length  $S$ , the number of clusters, containing the wild bootstrap weights, and  $\mathbf{K}$  is an  $N \times S$  matrix with elements  $k_{i,j} = 1$  if observation  $i$ ,  $i = 1, \dots, N$ , is in group  $j$  and 0 otherwise. The operator  $\odot$  is the Hadamard product that performs elementwise multiplication. By default,  $w_j = 1$  with probability 0.5 and  $w_j = -1$  with probability 0.5, that is, the rademacher error weights.

Alternatively, use one of the following error weights  $w$ :

- **mammen**, which is  $1 - \phi$  with probability  $\phi/\sqrt{5}$  and  $\phi$  otherwise, where  $\phi = (1 + \sqrt{5})/2$ .
  - **webb**, which takes the values  $-\sqrt{3/2}$ ,  $-\sqrt{2/2}$ ,  $-\sqrt{1/2}$ ,  $\sqrt{1/2}$ ,  $\sqrt{2/2}$ , and  $\sqrt{3/2}$ , each with probability  $1/6$ .
  - **normal**, which is a normal distribution with the first four moments given by 0, 1, 0, and 3.
  - **gamma**, which is a gamma distribution with shape parameter 4 and scale parameter  $1/2$ .
3. For each bootstrap sample, compute the unrestricted model and the null hypothesis  $t$  statistic,  $t^*$ . Include the observed  $t$  statistic,  $t$  in this set.
4. Given the  $B$  bootstrapped  $t$  statistics  $t_b^*$ ,  $b = 1, \dots, B$ , compute the wild bootstrap lower-tail  $p$ -value

$$P_l(t) = \frac{1}{B} \sum_{b=1}^B I(|t| < |t_b^*|)$$

and the upper-tail  $p$ -value

$$P_u(t) = \frac{1}{B} \sum_{b=1}^B I(|t| > |t_b^*|)$$

The reported equal-tail  $p$ -value is  $P_{et}(t) = 2 \min\{P_l(t), P_u(t)\}$ .

5. Compute the wild bootstrap confidence interval by searching for the lower limit and upper limit separately using the bisection root-finding algorithm. In doing so, find the quantities  $\text{ATET} = c_l$  and  $\text{ATET} = c_u$ ,  $c_l < c_u$ , such that  $(100 - L)/2\%$  of the bootstrapped distribution of the  $t$  statistics testing  $\text{ATET} = c$  is contained in either the lower or the upper tail. By default,  $L = 95$ .

The search algorithm does not always generate an  $L\%$  confidence interval. One reason for this situation is if  $B(100 - L)/200$  is not an integer, then there is not a slot in the sorted vector  $\mathbf{t}^*$  that corresponds to the upper- or lower-tail area  $(100 - L)/200$ . See [Roodman et al. \(2019, 8\)](#) for a good discussion.

Also, although a low-probability event, if at least one of the computed  $\mathbf{t}^*$  is (numerically) equal to the  $t$  testing  $\text{ATET} = c$ , or if there are ties between other computed values in  $\mathbf{t}^*$  adjacent to the  $t$  testing  $\text{ATET} = c$ , then the algorithm will not achieve the optimal coverage. In this case, we choose the confidence interval that is more conservative than requested. The achieved confidence level is reported in the ATET coefficient table.

The chance of ties in  $\mathbf{t}^*$  increases if the number of clusters,  $S$ , is small and the rademacher error weights are used, because there are  $2^S$  possible combinations of the two values 1 and  $-1$  in the weight vector  $\mathbf{w}$ . For example, if  $S = 10$ , then there are 1,024 possible combinations.

## Bias-corrected clustered standard error

Let there be  $S$  clusters,  $s = 1, \dots, S$ , each with  $N_s$  observations, and let  $N$  be the number of observations in the data. Let  $\mathbf{X}_s$  be the covariate matrix for cluster  $s$ , and let  $\mathbf{X}$  be the covariate matrix for all observations with dimension  $N \times L$ . Also, define  $\mathbf{I}_{N_s}$  to be an identity matrix of dimension  $N_s$ , and define  $\mathbf{P}_{ss} = \mathbf{X}_s (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'_s$  as a projection matrix for cluster  $s$ . Finally, let  $\hat{\varepsilon}_s$  be the residuals corresponding to cluster  $s$ . The bias-corrected cluster variance–covariance matrix is given by the following:

$$\mathbf{V} = \left( \sum_{s=1}^S \mathbf{X}'_s \mathbf{X}_s \right)^{-1} \sum_{s=1}^S \mathbf{X}'_s (\mathbf{I}_{N_s} - \mathbf{P}_{ss})^{-1/2} \hat{\varepsilon}_s \hat{\varepsilon}'_s \left\{ (\mathbf{I}_{N_s} - \mathbf{P}_{ss})^{-1/2} \right\}' \mathbf{X}_s \left( \sum_{s=1}^S \mathbf{X}'_s \mathbf{X}_s \right)^{-1} \quad (6)$$

Let  $\mathbf{P}$  be the projection matrix for all the data, and let  $(\mathbf{I}_N - \mathbf{P})_s$  be an  $N_s \times N$  matrix of rows of the  $N \times N$  matrix  $(\mathbf{I}_N - \mathbf{P})$ . Finally, let  $\mathbf{e}_{L,k}$  be an  $L$  vector with  $k$ th component equal to 1 and equal to 0 elsewhere. We define the matrix  $\mathbf{G}$  as the matrix with  $s$ th column given by

$$\mathbf{G}_s = (\mathbf{I}_N - \mathbf{P})'_s (\mathbf{I}_{N_s} - \mathbf{P}_{ss})^{-1/2} \mathbf{X}_s (\mathbf{X}'\mathbf{X})^{-1} \mathbf{e}_{L,k}$$

We compute the degrees-of-freedom adjustment proposed by [Bell and McCaffrey \(2002\)](#),  $K_{\text{BM}}$ , as follows,

$$\begin{aligned} K_{\text{BM}} &= \frac{\text{tr}(\mathbf{G}'\mathbf{G})^2}{\text{tr}\left\{(\mathbf{G}'\mathbf{G})^2\right\}} \\ &= \frac{\left(\sum_{i=1}^S \lambda_i\right)^2}{\sum_{i=1}^S \lambda_i^2} \end{aligned}$$

where  $\text{tr}(\cdot)$  is the trace function and  $\lambda_i$  is the  $i$ th eigenvalue of the matrix  $\mathbf{G}'\mathbf{G}$ .

When  $S$  is large, computing the eigenvalues can be time consuming. We define

$$\begin{aligned} \mathbf{a}_s &= (\mathbf{I}_{N_s} - \mathbf{P}_{ss})^{-\frac{1}{2}} \mathbf{X}_s (\mathbf{X}'\mathbf{X})^{-1} \mathbf{e}_{L,k} \\ \mathbf{b}_s &= \mathbf{P}_s \mathbf{a}_s \\ \mathbf{A} &= (\mathbf{a}'_1 \mathbf{a}_1, \dots, \mathbf{a}'_S \mathbf{a}_S)' = (A_1, A_2, \dots, A_S)' \\ \mathbf{B} &= (\mathbf{b}_1, \dots, \mathbf{b}_S) = \begin{pmatrix} B_{1,1} & B_{1,2} & \cdots & B_{1,S} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N,1} & B_{N,2} & \cdots & B_{N,S} \end{pmatrix} \end{aligned}$$

where  $\mathbf{P}_s = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'_s$  and  $B_{i,s}$ ,  $i = 1, \dots, N$ , are the elements of the vectors  $\mathbf{b}_s$ ,  $s = 1, \dots, S$ . Then  $\mathbf{G}'\mathbf{G} = \text{diag}(\mathbf{A}) - \mathbf{B}'\mathbf{B}$ . We now express the adjusted degrees of freedom ([Kolesár 2021](#)) as

$$K_{\text{BM}} = \frac{\left(\sum_{s=1}^S A_s - \sum_{i=1}^N \sum_{s=1}^S B_{i,s}^2\right)^2}{\sum_{s=1}^S A_s^2 - 2 \sum_{s=1}^S A_s \sum_{i=1}^N B_{i,s}^2 + \sum_{s_1=1}^S \sum_{s_2=1}^S (\mathbf{b}'_{s_1} \mathbf{b}_{s_2})^2}$$

which can be computed efficiently in Mata and using QR decomposition. For example, by decomposing  $\mathbf{X} = \mathbf{QR}$ , where  $\mathbf{Q}$  is  $n \times k$  and orthonormal and  $\mathbf{R}$  is  $k \times k$  and upper triangular. The matrix  $\mathbf{B}$  can be rewritten so that it has dimension  $k \times S$  instead of  $N \times S$ . We can do even better if the column rank of  $\mathbf{X}$  is  $r \leq k$  by taking advantage of QR column pivoting.

When weights are specified, we use the weighted-covariate matrix  $\tilde{\mathbf{X}} = \text{diag}(\mathbf{w})^{\frac{1}{2}} \mathbf{X}$  and its corresponding projection matrix  $\tilde{\mathbf{P}}$ , as well as the cluster covariance matrices  $\tilde{\mathbf{X}}_s$ , their projection matrices  $\tilde{\mathbf{P}}_{ss}$ , and weighted residuals  $\tilde{\varepsilon}_s = \text{diag}(\mathbf{w}_s)^{\frac{1}{2}} \hat{\varepsilon}_s$ .

For the ATET coefficient, we use the standard errors from (6) and the degrees of freedom in  $K_{\text{BM}}$  to construct confidence intervals.

## Acknowledgment

The algorithm used to compute the wild bootstrap was made faster using techniques described in Roodman et al. (2019). We thank David Roodman for bringing this to our attention.

## References

- Abadie, A., and M. D. Cattaneo. 2018. Econometric methods for program evaluation. *Annual Review of Economics* 10: 465–503. <https://doi.org/10.1146/annurev-economics-080217-053402>.
- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.
- . 2015. *Mastering Metrics: The Path from Cause to Effect*. Princeton, NJ: Princeton University Press.
- Bell, R. M., and D. F. McCaffrey. 2002. Bias reduction in standard errors for linear regression with multi-stage samples. *Survey Methodology* 28: 169–181.
- Bertrand, M., E. Duflo, and S. Mullainathan. 2004. How much should we trust difference-in-differences estimates? *Quarterly Journal of Economics* 119: 249–275. <https://doi.org/10.1162/003355304772839588>.
- Blundell, R. W., and M. C. Dias. 2009. Alternative approaches to evaluation in empirical microeconomics. *Journal of Human Resources* 44: 565–640. <https://doi.org/10.1007/s10258-002-0010-3>.
- Cameron, A. C., J. B. Gelbach, and D. L. Miller. 2008. Bootstrap-based improvements for inference with clustered errors. *Review of Economics and Statistics* 90: 414–427. <https://doi.org/10.1162/rest.90.3.414>.
- Cameron, A. C., and D. L. Miller. 2015. A practitioner's guide to cluster-robust inference. *Journal of Human Resources* 50: 317–372. <https://doi.org/10.3368/jhr.50.2.317>.
- Clarke, D., and K. Tapia-Schythe. 2021. Implementing the panel event study. *Stata Journal* 21: 853–884.
- Donald, S. G., and K. Lang. 2007. Inference with difference-in-differences and other panel data. *Review of Economics and Statistics* 89: 221–233. <https://doi.org/10.1162/rest.89.2.221>.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.
- Kolesár, M. 2021. Robust standard errors in small samples. <https://cran.r-project.org/web/packages/dfadjust/vignettes/dfadjust.pdf>.
- Lechner, M. 2011. The estimation of causal effects by difference-in-difference methods. *Foundations and Trends in Econometrics* 4: 165–224. <http://doi.org/10.1561/08000000014>.
- MacKinnon, J. G. 2019. How cluster-robust inference is changing applied econometrics. *Canadian Journal of Economics* 52: 851–881. <https://doi.org/10.1111/caje.12388>.
- MacKinnon, J. G., and M. D. Webb. 2018. The wild bootstrap for few (treated) clusters. *Econometrics Journal* 21: 114–135. <https://doi.org/10.1111/ectj.12107>.
- Moser, P., and A. Voena. 2012. Compulsory licensing: Evidence from the Trading with the Enemy Act. *American Economic Review* 102: 396–427. <https://doi.org/10.1257/aer.102.1.396>.

- Roodman, D., J. G. MacKinnon, M. O. Nielsen, and M. D. Webb. 2019. *Fast and wild: Bootstrap inference in Stata using boottest*. *Stata Journal* 19: 4–60.
- Schmidheiny, K., and S. Siegloch. 2019. On event study designs and distributed-lag models: Equivalence, generalization and practical implications. IZA Discussion Papers No. 12079, Institute of Labor Economics (IZA). <https://docs.iza.org/dp12079.pdf>.
- Wing, C., K. Simon, and R. A. Bello-Gomez. 2018. Designing difference in difference studies: Best practices for public health policy research. *Annual Review of Public Health* 39: 453–469. <https://doi.org/10.1146/annurev-publhealth-040617-013507>.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

## Also see

- [TE] **didregress postestimation** — Postestimation tools for didregress and xtdidregress
- [TE] **DID intro** — Introduction to difference-in-differences estimation
- [TE] **teffects intro** — Introduction to treatment effects for observational data
- [TE] **teffects intro advanced** — Advanced introduction to treatment effects for observational data
- [U] **20 Estimation and postestimation commands**



[Postestimation commands](#)      [predict](#)      [estat](#)  
[Methods and formulas](#)      [Also see](#)

## Postestimation commands

The following postestimation commands are of special interest after `didregress` and `xt-didregress`:

Command	Description
<code>estat trendplots</code>	graphical diagnostics for parallel trends
<code>estat ptrends</code>	parallel-trends test
<code>estat granger</code>	Granger causality test
<code>estat grangerplot</code>	time-specific treatment effects

The following standard postestimation commands are also available:

Command	Description
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
<code>forecast</code>	dynamic forecasts and simulations
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of coefficients
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
<code>predict</code>	linear predictions and residuals
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

# predict

## Description for predict

`predict` creates a new variable containing predictions such as linear predictors and residuals.

## Menu for predict

Statistics > Postestimation

## Syntax for predict

```
predict [type] newvar [if] [in] [, statistic]
```

<i>statistic</i>	Description
------------------	-------------

---

Main

<code>xb</code>	linear predictor; the default
<code>residuals</code>	residuals

---

## Options for predict

Main

`xb`, the default, calculates the linear predictor. It excludes the effect of the first group or of the panel identifier. All other effects, including the time fixed effects, are included in the linear predictor.

`residuals` calculates the overall residuals. It is the difference of the outcome and the linear predictor, including all group, panel, and time effects. In other words, it is not just the difference of the outcome and the linear predictor.

## estat

### Description for estat

`estat trendplots` produces two diagnostic plots for assessing the parallel-trends assumption that is required for consistent estimation of the ATET using `didregress` or `xtdidregress`. The first plot consists of two lines showing the mean of the outcome over time for the treatment and the control groups. The second plot augments the DID model to include interactions of time with an indicator of treatment and plots the predicted values of this augmented model for the treatment and control groups. Both plots include a vertical line one period before treatment.

`estat ptrends` performs a test of whether the linear trends in the outcome variable are parallel between control and treatment groups during the pretreatment period.

`estat granger` performs a test of whether treatment effects can be observed in anticipation of the treatment.

`estat grangerplot` produces a graph of time-specific treatment effects by plotting coefficients from leads and lags of the treatment indicator variable.

### Menu for estat

Statistics > Postestimation

### Syntax for estat

*Graphical diagnostics for parallel trends*

```
estat trendplots [ , trend_options plot_options ]
```

*Parallel-trends test*

```
estat ptrends
```

*Granger causality test*

```
estat granger
```

*Time-specific treatment effects*

```
estat grangerplot [ , grangerplot_options ]
```

*trend\_options*

Description

---

<code>omeans</code>	draw graph showing observed means
<code>omeans(plot_options)</code>	draw observed-means graph and affect its rendition
<code>ltrends</code>	draw graph showing linear trends
<code>ltrends(plot_options)</code>	draw linear-trends graph and affect its rendition
<code>notitle</code>	suppress overall title
<code>noxline</code>	suppress treatment-time reference line
<code>nocommonlegend</code>	display two individual legends
<code>legendfrom(#)</code>	specify which legend to use

---

<i>plot_options</i>	Description
Plot	
<i>cline_options</i>	affect rendition of the plotted trend lines; see [G-3] <i>cline_options</i>
<i>line1opts(cline_options)</i>	affect rendition of the line for controls
<i>line2opts(cline_options)</i>	affect rendition of the line for treated
Y axis, X axis, Titles, Legend, Overall	
<i>twoway_options</i>	any options other than <code>by()</code> documented in [G-3] <i>twoway_options</i>
<i>grangerplot_options</i>	Description
<i>nleads(#)</i>	number of leads
<i>nlags(#)</i>	number of lags
<i>baseline(#)</i>	baseline period
<i>lagview</i>	show lags instead of time units
<i>verbose</i>	display results of the underlying regression
<i>post</i>	post the results of the underlying regression in $e()$
<i>noci</i>	do not plot confidence intervals
<i>noyline</i>	suppress $y$ -axis reference line
CI plot	
<i>recastci(plotype)</i>	plot confidence intervals using <i>plotype</i>
<i>ciopts(rcap_options)</i>	affect rendition of confidence intervals
<i>level(#)</i>	set confidence level; default is <code>level(95)</code>
Add plots	
<i>addplot(plot)</i>	add other plots to the graph
Y axis, X axis, Titles, Legend, Overall	
<i>twoway_options</i>	any options other than <code>by()</code> documented in [G-3] <i>twoway_options</i>

## Options for estat trendplots

`omeans`, `omeans(plot_options)`, `ltrends`, and `ltrends(plot_options)` specify which graphs are to be included and how they should be individually rendered. The default is `omeans ltrends`, meaning that both graphs are included without any modifications.

`omeans` specifies that the observed-means graph be included. Specifying `omeans` suppresses the linear-trends model graph unless `ltrends` or `ltrends(plot_options)` is also specified.

`omeans(plot_options)` specifies that the observed-means graph be included and affects its rendition. Specifying `omeans(plot_options)` suppresses the linear-trends model graph unless `ltrends` or `ltrends(plot_options)` is also specified.

`ltrends` specifies that the linear-trends model graph be included. Specifying `ltrends` suppresses the observed-means graph unless `omeans` or `omeans(plot_options)` is also specified.

`ltrends(plot_options)` specifies that the linear-trends model graph be included and affects its rendition. Specifying `ltrends(plot_options)` suppresses the observed-means graph unless `omeans` or `omeans(plot_options)` is also specified.

`notitle` suppresses the overall title of the rendered graph.

`noxline` suppresses rendering of the treatment-time reference line.

`nocommonlegend` suppresses the display of one common legend and renders two individual legends.

`legendfrom(#)` specifies which legend to use; the default is `legendfrom(1)`, which refers to the legend of the first plot (observed means). `legendfrom(#)` is not allowed with the `nocommonlegend` option.

#### Plot

`cline_options` affect the rendition of the plotted trend lines, including their style, size, and color; see [G-3] [cline\\_options](#).

`line1opts(cline_options)` affect the rendition of the plotted trend lines for the group of controls, including their style, size, and color; see [G-3] [cline\\_options](#).

`line2opts(cline_options)` affect the rendition of the plotted trend lines for the group of treated, including their style, size, and color; see [G-3] [cline\\_options](#).

#### Y axis, X axis, Titles, Legend, Overall

`twoway_options` are any of the options documented in [G-3] [twoway\\_options](#), excluding `by()`. These include options for titling the graph (see [G-3] [title\\_options](#)) and for saving the graph to disk (see [G-3] [saving\\_option](#)).

## Options for `estat grangerplot`

`nleads(#)` specifies the number of leads to be included in the model and plotted. By default, all available leads are included. The number of leads must be greater than 0.

`nlags(#)` specifies the number of lags to be included in the model and plotted. By default, all available lags are included. The number of lags must be greater than or equal to 0.

`baseline(#)` specifies the baseline period for which the corresponding lead or lag is omitted. By default, the first lead is omitted, which corresponds to the time period prior to intervention.

`lagview` specifies to show the values of the  $x$  axis in terms of lags. If this option is not specified, time values are shown.

`verbose` specifies to display the output of the underlying regression model.

`post` posts the results of the underlying regression as estimation results in `e()`.

`noci` removes the pointwise confidence intervals. The default is to plot the confidence intervals.

`noyline` suppresses rendering of the reference line.

#### CI plot

`recastci(plottype)` specifies that confidence intervals be plotted using *plottype*. *plottype* may be `rarea`, `rbar`, `rspike`, `rcap`, `rcapsym`, `rline`, `rconnected`, or `rscatter`; see [G-2] [graph twoway](#). When `recastci()` is specified, the plot-rendition options appropriate to the specified *plottype* may be used in lieu of `rcap_options`. For details on those options, follow the appropriate link from [G-2] [graph twoway](#).

`ciopts(rcap_options)` affects the rendition of confidence intervals; see [G-3] [rcap\\_options](#).

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see [U] [20.8 Specifying the width of confidence intervals](#).

Add plots

`addplot(plot)` provides a way to add other plots to the generated graph; see [G-3] [addplot\\_option](#).

Y axis, X axis, Titles, Legend, Overall

`twoway_options` are any of the options documented in [G-3] [twoway\\_options](#), excluding `by()`. These include options for titling the graph (see [G-3] [title\\_options](#)) and for saving the graph to disk (see [G-3] [saving\\_option](#)).

## Stored results for `estat`

`estat ptrends` stores the following results for the test of linear trends in `r()`:

Scalars

<code>r(N)</code>	number of observations
<code>r(F)</code>	test statistic
<code>r(df_r)</code>	number of degrees of freedom of the residuals for the $F$ distribution under $H_0$
<code>r(p)</code>	$p$ -value
<code>r(df_m)</code>	number of degrees of freedom of the test for the $F$ distribution under $H_0$

`estat granger` stores the following results for the test of treatment anticipation in `r()`:

Scalars

<code>r(N)</code>	number of observations
<code>r(F)</code>	test statistic
<code>r(df_r)</code>	number of degrees of freedom of the residuals for the $F$ distribution under $H_0$
<code>r(p)</code>	$p$ -value
<code>r(df_m)</code>	number of degrees of freedom of the test for the $F$ distribution under $H_0$

`estat grangerplot`, when used with option `post`, stores results from the underlying regression model in `e()` and `r()`.

## Methods and formulas

The tests performed with `estat ptrends` and `estat granger` are based on augmented difference-in-differences (DID) models. With `estat ptrends`, we augment the DID model with terms that capture the differences in slopes between treated and controls. With `estat granger`, we augment the model by interacting the dummy variable that marks treated observations with dummy variables for time periods prior to the treatment to capture any potential anticipatory treatment effects.

Let's consider the case of panel data for individuals over time, in which individuals belong to a group  $s$ . Groups could be states, occupational categories, districts, etc. Let  $y_{it,s}$  be the outcome of individual  $i$ , who belongs to group  $s$ , at time  $t$ , where  $i = 1, \dots, N$ ,  $t = 1, \dots, T$ , and  $s = 1, \dots, S$ .

We can write the DID model for such setups as follows:

$$y_{ist} = \gamma_i + \gamma_t + \mathbf{x}_{ist}\beta + D_{st}\delta + \epsilon_{ist}$$

Here  $\gamma_i$  are individual fixed effects,  $\gamma_t$  are time fixed effects,  $\mathbf{x}_{ist}$  are covariates,  $D_{st}$  is a variable that is 1 if an individual belongs to a group  $s$  that is treated at time  $t$  and is 0 otherwise, and  $\epsilon_{ist}$  is an error term. The coefficient  $\delta$  represents the average treatment effect on the treated (ATET).

To simplify the exposition below, we rewrite the model as follows:

$$\begin{aligned} y_{ist} &= \gamma_i + \gamma_t + \mathbf{x}_{ist}\boldsymbol{\beta} + D_{st}\delta + \epsilon_{ist} \\ y_{ist} &= \text{DID}_{ist} + \epsilon_{ist} \end{aligned} \tag{1}$$

The linear-trends model that is used for the parallel-trends test with `estat ptrends` augments the above model with two more terms. Let  $d_{t,0} = 1(d_t = 0)$  be a variable indicating pretreatment time periods, and let  $d_{t,1} = 1(d_t = 1)$  be a variable indicating posttreatment time periods. Also, let  $w_i$  be a variable that is 1 if the individual belongs to a treated group and is 0 otherwise. The augmentation terms then consist of two 3-way interactions between  $d_{t,0}$ ,  $w_i$ , and  $t$ , and  $d_{t,1}$ ,  $w_i$ , and  $t$ :

$$y_{ist} = \text{DID}_{ist} + w_i d_{t,0} t \zeta_1 + w_i d_{t,1} t \zeta_2 + \epsilon_{ist} \tag{2}$$

Under this specification, the coefficient  $\zeta_1$  captures the differences in slopes between treatment group and control group in pretreatment periods, while  $\zeta_2$  captures the differences in slopes in posttreatment periods. If  $\zeta_1$  is 0, the linear trends in the outcome are parallel during pretreatment periods. The same is true for  $\zeta_2$  with respect to the posttreatment period; however, posttreatment differences in trends are not relevant for assessing the parallel-trends assumption. `estat ptrends` uses a Wald test of  $\zeta_1$  against 0 to assess whether the linear trends are parallel prior to treatment. Thus, the null hypothesis of this test is that the linear trends are parallel.

`estat granger` performs a Granger-type causality test to assess whether treatment effects are observed prior to the treatment. To illustrate this, suppose the treatment took place at time  $t = j$ . We could express  $D_{st}$  as  $D_{st} = 1(t \geq j)w_i$ . The Granger-type test augments the model with counterfactual treatment-time indicators. For example, if the treatment occurred at time  $j - 1$ , then we could construct a new treatment as  $1(t_{it} \geq j - 1)w_i$ , and if we have sufficient time points, we could construct another counterfactual treatment as  $1(t_{it} \geq j - 2)w_i$ , and so on. These terms are referred to as leads in the DID literature. The model used by `estat granger` uses the model in (1) and augments it with all leads leaving out one for identification purposes. Let  $J$  index the time at which the treatment occurs.

$$y_{ist} = \text{DID}_{ist} + \sum_{j=2}^{J-1} 1(t_{it} \geq j)w_i \lambda_j + \nu_{ist} \tag{3}$$

The test result is then obtained by performing a joint Wald test on the coefficients  $\lambda_j$ . Thus, the null hypothesis for this test is that the coefficients in  $\lambda_j$  are jointly 0, which is to say there are no anticipatory effects.

`estat grangerplot` fits a generalization of the DID model in (1) and plots the estimated coefficients from this model, including their 95% confidence intervals. The model is similar to (3), but this model parameterizes the leads differently and includes lags in addition to leads. Let  $I_s$  be the time of treatment,  $m < 0$  be the number of time periods prior to  $I_s$ ,  $q \geq 0$  be the number of periods after  $I_s$ , and  $b$  be the baseline period. The model is

$$y_{ist} = \gamma_i + \gamma_t + \mathbf{x}_{ist}\boldsymbol{\beta} + \sum_{k=m, k \neq b}^q B_{st}^k w_i \lambda_k + \epsilon_{ist}$$

where

$$B_{st}^k = \begin{cases} 1(t_{it} \leq I_s + k), & \text{if } k = m \\ 1(t_{it} = I_s + k), & \text{if } m < k < q \\ 1(t_{it} \geq I_s + k), & \text{if } k = q \end{cases}$$

This yields a model with  $|m|$  leads and  $q$  lags. By default, `estat grangerplot` uses all available leads and lags. If, without loss of generality, we set the base to the period prior to treatment,  $b = -1$  (the default), then with  $t = 1, \dots, T$  and  $I_s = J$ , a maximum of  $n_{\text{leads}} = J - 2$  leads and  $n_{\text{lags}} = T - J + 1$  lags is available. Notice that, if all available leads and lags are used,  $B_{st}^k$  reduces to  $B_{st}^k = 1(t_{it} = I_s + k)$  because there are no periods before or after  $I_s + k$ . With fewer than available leads and lags, that is,  $|m| < n_{\text{leads}}$  or  $q < n_{\text{lags}}$ , notice that the corresponding indicator variables capture the periods beyond the endpoints that correspond to  $m$  and  $q$ .

At a minimum, the model has to include a single lead. In that case, we have that  $m = -1$  and  $q = 0$ . After omitting the base, we have that  $k = 0$  and  $B_{st}^k$  reduces to  $B_{st}^k = 1(t_{it} \geq I_s)$ . Notice that  $B_{st}^k w_i$  now yields our original treatment indicator  $D_{st}$ . In other words, the model with a single lead and no lags yields the special case of the DID model in (1). Notice also that the model in (3) is a special case, too. It is equivalent to the model fit by `estat grangerplot` with all available leads and no lags. However, (3) uses a different parameterization because the indicator variables are constructed differently. `estat grangerplot` plots the coefficients  $\lambda_k$  against the corresponding time periods.

The `estat trendplots` command produces two plots. The first plot is simply plotting the observed means for each treatment group at each point in time. The second plot is based on the model in (2), which is the model used for the parallel-trends test, but this model centers the continuous time variable around its minimum value:

$$y_{ist} = \text{DID}_{ist} + w_i d_{t,0} \{t - \min(t)\} \zeta_1 + w_i d_{t,1} \{t - \min(t)\} \zeta_2 + \mu_{ist}$$

Centering around the minimum time value provides a common reference point at the first observed time point such that deviations from parallelism are easily detectable. The graph then shows the predicted values from this model, evaluated at all observed time points for each of the treatment groups and at the means of the covariates.

While the formulas above are shown for the case of panel data, these methods work the same way for data that consist of repeated cross-sections.

## Also see

[TE] **didregress** — Difference-in-differences estimation

[TE] **DID intro** — Introduction to difference-in-differences estimation

[U] **20 Estimation and postestimation commands**



# Title

**eteffects** — Endogenous treatment-effects estimation

[Description](#)

[Options](#)

[Acknowledgment](#)

[Quick start](#)

[Remarks and examples](#)

[References](#)

[Menu](#)

[Stored results](#)

[Also see](#)

[Syntax](#)

[Methods and formulas](#)

## Description

`eteffects` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational data when treatment assignment is correlated with the potential outcomes. It allows for continuous, binary, count, fractional, and nonnegative outcomes and requires a binary treatment. To control for the endogeneity of the treatment assignment, the estimator includes residuals from the treatment model in the models for the potential outcomes, known as a control-function approach.

## Quick start

ATE of binary treatment `treat` using a linear model for outcome `y1` on `x` and the residuals from a probit model for `treat` on `x` and `z`

```
eteffects (y1 x) (treat x z)
```

As above, but estimate ATET

```
eteffects (y1 x) (treat x z), atet
```

As above, but estimate POMs

```
eteffects (y1 x) (treat x z), pomeans
```

As above, and show parameters from auxiliary equations

```
eteffects (y1 x) (treat x z), pomeans aequations
```

ATE of `treat` using an exponential-mean model for `y1`

```
eteffects (y1 x, exponential) (treat x z)
```

Same as above, but for count outcome `y2`

```
eteffects (y2 x, exponential) (treat x z)
```

As above, but use a probit model for binary outcome `y3`

```
eteffects (y3 x, probit) (treat x z)
```

As above, but use a fractional probit model for `y4` ranging from 0 to 1

```
eteffects (y4 x, fractional) (treat x z)
```

## Menu

Statistics > Treatment effects > Endogenous treatment > Control function estimator > Continuous outcomes

Statistics > Treatment effects > Endogenous treatment > Control function estimator > Binary outcomes

Statistics > Treatment effects > Endogenous treatment > Control function estimator > Count outcomes

Statistics > Treatment effects > Endogenous treatment > Control function estimator > Fractional outcomes

Statistics > Treatment effects > Endogenous treatment > Control function estimator > Nonnegative outcomes

## Syntax

```
eteffects (ovar omvarlist [, omodel noconstant])
         (tvar tmvarlist [, noconstant]) [if] [in] [weight] [, stat options]
```

*ovar* is the *depvar* of the outcome model.

*omvarlist* is the list of exogenous *indepvars* in the outcome model.

*tvar* is the binary treatment variable.

*tmvarlist* is the list of covariates that predict treatment assignment.

---

<i>omodel</i>	Description
---------------	-------------

---

Model	
-------	--

<b>linear</b>	linear outcome model; the default
<b>fractional</b>	fractional probit outcome model
<b>probit</b>	probit outcome model
<b>exponential</b>	exponential-mean outcome model

---



---

<i>stat</i>	Description
-------------	-------------

---

Model	
-------	--

<b>ate</b>	estimate average treatment effect in population; the default
<b>atet</b>	estimate average treatment effect on the treated
<b>pomeans</b>	estimate potential-outcome means

---



---

<i>options</i>	Description
----------------	-------------

---

Model	
-------	--

<b>noconstant</b>	suppress constant term
-------------------	------------------------

SE/Robust	
-----------	--

<b>vce(<i>vcetype</i>)</b>	<i>vcetype</i> may be <b>robust</b> , <b>cluster <i>clustvar</i></b> , <b>bootstrap</b> , or <b>jackknife</b>
----------------------------	---

Reporting	
-----------	--

<b>level(#)</b>	set confidence level; default is level(95)
<b>aequations</b>	display auxiliary-equation results
<b>display_options</b>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

Maximization	
--------------	--

<b>maximize_options</b>	control the maximization process; seldom used
-------------------------	---

Advanced	
----------	--

<b>pstolerance(#)</b>	set tolerance for overlap assumption
<b>osample(<i>newvar</i>)</b>	generate <i>newvar</i> to mark observations that violate the overlap assumption
<b>coeflegend</b>	display legend instead of statistics

---

*omvarlist* and *tmvarlist* may contain factor variables; see [U] 11.4.3 Factor variables.

*bootstrap*, *by*, *collect*, *jackknife*, and *statsby* are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the *bootstrap* prefix; see [R] *bootstrap*.

*fweights*, *iwweights*, and *pweights* are allowed; see [U] 11.1.6 *weight*.

*coeflegend* does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

## Options

### Model

*noconstant*; see [R] Estimation options.

*stat* is one of three statistics: *ate*, *atet*, or *pomeans*. *ate* is the default.

*ate* specifies that the average treatment effect be estimated.

*atet* specifies that the average treatment effect on the treated be estimated.

*pomeans* specifies that the potential-outcome means for each treatment level be estimated.

### SE/Robust

*vce(vcetype)* specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (*robust*), that allow for intragroup correlation (*cluster clustvar*), and that use bootstrap or jackknife methods (*bootstrap*, *jackknife*); see [R] *vce\_option*.

### Reporting

*level(#)*; see [R] Estimation options.

*aequations* specifies that the results for the outcome-model or the treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

*display\_options*: *noci*, *nopvalues*, *noomitted*, *vsquish*, *noemptycells*, *baselevels*, *allbaselevels*, *nofvlabel*, *fvwrap(#)*, *fvwrapon(style)*, *cformat(%fmt)*, *pformat(%fmt)*, *sformat(%fmt)*, and *nolstretch*; see [R] Estimation options.

### Maximization

*maximize\_options*: *iterate(#)*, *[no]log*, and *from(init\_specs)*; see [R] Maximize. These options are seldom used.

*init\_specs* is one of

*matname* [ , *skip copy* ]

*#* [ , *# ...* ] , *copy*

### Advanced

*pstolerance(#)* specifies the tolerance used to check the overlap assumption. The default value is *pstolerance(1e-5)*. *eteffects* will exit with an error if an observation has an estimated propensity score smaller than that specified by *pstolerance()*.

*osample(newvar)* specifies that indicator variable *newvar* be created to identify observations that violate the overlap assumption.

The following option is available with *eteffects* but is not shown in the dialog box:

*coeflegend*; see [R] Estimation options.

## Remarks and examples

If you are unfamiliar with treatment-effects estimators for observational data or the `teffects` commands, we recommend that you look at [TE] [teffects intro](#). For the intuition behind some of the concepts discussed below, we recommend that you read *Defining treatment effects* in [TE] [teffects intro advanced](#).

The estimators implemented in `eteffects` extend the regression adjustment (RA) estimators implemented in `teffects ra` to allow for endogenous treatments, that is, when treatment assignment is not independent of outcomes. This endogeneity is a violation of the conditional mean independence assumption used by `teffects ra`, as discussed in *The potential-outcome model* in [TE] [teffects intro advanced](#).

`eteffects` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs). It uses a linear, a probit, a fractional probit, or an exponential-mean model for the potential outcomes and a probit model for treatment assignment. After conditioning on the observable covariates, `eteffects` allows some remaining unobservable components to affect both treatment assignment and the potential outcomes. The treatment assignment process is endogenous because these unobservable components affect both treatment assignment and the potential outcomes.

To control for the endogeneity of the treatment assignment, `eteffects` uses a control-function approach. This method controls for endogeneity by including the residuals from the treatment-assignment model as a regressor in the models for the potential outcome. The implementation in `eteffects` follows [Wooldridge \(2010\)](#), who provides an excellent discussion of the control-function approach that addresses endogeneity problems in a treatment-effects context.

The control-function approach estimates the parameters of the conditional means of the potential outcomes. Sample averages of the conditional means are used to estimate the unconditional ATE, ATET, or POMs. This method is known as RA.

Taken collectively, the estimators implemented in `eteffects` are control-function RA estimators. See [Methods and formulas](#) below for details about the estimation procedure.

### ► Example 1: Linear outcome estimates for ATE

Suppose we want to know the effect of a mother smoking while pregnant on the birthweight of her infant. We use an extract from [Cattaneo \(2010\)](#) in which `bweight` records the baby's birthweight and `mbsmoke` is the variable (0 or 1) indicating whether a mother smoked while pregnant.

We may believe that birthweight (the potential outcome) is influenced by whether the mother had a prenatal exam in the first trimester, whether the mother is married, the mother's age, whether this is the first birth, and the education level of the father. We may also believe that the smoking decision (the treatment) is influenced by the mother's marital status, the education level of the mother, her age, whether she had a prenatal exam in the first trimester, and whether this baby is her first baby.

Thus we condition on different sets of covariates in the models for treatment assignment and the potential outcomes. In the probit model for smoking status (`mbsmoke`), we condition on marital status (`mmarried`), age (`mage`), mother's education level (`medu`), father's education level (`fedu`), and whether it was the mother's first baby (`fbaby`). We model birthweight (`bweight`) as a linear function of whether the mother had a first-trimester prenatal exam (`prenatal1`), `mmarried`, `mage`, and `fbaby`. We can estimate the ATE of smoking status using one of the `teffects` estimators if we believe that there are no unobservable components that affect both the decision to smoke while pregnant and the potential birthweights.

If we believe there is some unobservable factor that affects both assignment to treatment and the potential outcome, we must select another estimator. For example, we do not observe a mother's health consciousness, which affects both the smoking decision and each potential birthweight through other behaviors such as intake of prenatal vitamins. Under these assumptions, the estimators in `eteffects` consistently estimate the ATE, but the estimators in `[TE] teffects` yield inconsistent estimates.

```
. use https://www.stata-press.com/data/r17/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)

. eteffects (bweight i.prenatal1 i.mmarrried mage i.fbaby)
> (mbsmoke i.mmarrried mage i.fbaby medu fedu)

Iteration 0: EE criterion = 4.704e-24
Iteration 1: EE criterion = 1.223e-25

Endogenous treatment-effects estimation          Number of obs = 4,642
Outcome model: linear
Treatment model: probit
```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
mbsmoke (Smoker vs Nonsmoker)	-455.9119	212.4393	-2.15	0.032	-872.2853	-39.53852
POmean						
mbsmoke Nonsmoker	3437.964	31.21145	110.15	0.000	3376.791	3499.138

When no mother smokes, the average birthweight is 3,438 grams. The average birthweight is 456 grams less when all mothers smoke than when no mother smokes.

We can compare these results with those obtained if we ignore the endogeneity of the smoking decision. Below we estimate the ATE using the inverse-probability-weighted regression-adjustment estimator in `[TE] teffects ipwra`.

```
. teffects ipwra (bweight i.prenatal1 i.mmarrried mage i.fbaby)
> (mbsmoke i.mmarrried mage i.fbaby medu fedu)

Iteration 0: EE criterion = 3.036e-22
Iteration 1: EE criterion = 3.755e-26

Treatment-effects estimation          Number of obs    =    4,642
Estimator      : IPW regression adjustment
Outcome model  : linear
Treatment model: logit
```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
mbsmoke (Smoker vs Nonsmoker)	-233.6835	25.07695	-9.32	0.000	-282.8335	-184.5336
POmean						
mbsmoke Nonsmoker	3403.191	9.529709	357.11	0.000	3384.513	3421.869

In magnitude, the estimated ATE is more than half the estimate that allows for endogenous treatment assignment. If there is endogeneity, disregarding it underestimates the effect of smoking on birthweight. We show how to test for endogeneity in [example 1](#) of [\[TE\] eteffects postestimation](#).

◀

### ▷ Example 2: Estimating the ATET

Continuing [example 1](#), we can use the `atet` option to estimate the ATET.

```
. eteffects (bweight i.prenatal1 i.mmarried mage i.fbaby)
> (mbsmoke i.mmarried mage i.fbaby medu fedu), atet

Iteration 0:   EE criterion = 4.688e-24
Iteration 1:   EE criterion = 8.479e-26

Endogenous treatment-effects estimation           Number of obs = 4,642
Outcome model: linear
Treatment model: probit
```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATET						
mbsmoke (Smoker vs Nonsmoker)	-409.8527	161.4816	-2.54	0.011	-726.3507	-93.35466
POmean						
mbsmoke Nonsmoker	3547.512	160.0595	22.16	0.000	3233.801	3861.223

In the population of mothers who smoke, the average infant birthweight would be 3,548 grams if none of these mothers smoked. For the mothers who smoke, the average infant birthweight is 410 grams less than if none of these mothers smoked.

◀

### ▷ Example 3: Exponential-mean outcomes

We estimate the ATE of living in an urban area on monthly earnings (`wage`), using a subset of the National Longitudinal Survey in 1980 found in [Wooldridge \(2010\)](#). We assume that once we condition on work experience (`exper`), whether education level attained is college or higher (`college`), and IQ (`iq`), individual wages follow an exponential mean. The variables used to predict residence in an urban area (`urban`) are `college` and whether the respondent's father attained a bachelor's degree or higher (`fcollege`).

```
. use https://www.stata-press.com/data/r17/nlsy80
. eteffects (wage exper iq i.college, exponential nocons)
> (urban i.college fcollege)
Iteration 0: EE criterion = 2.903e-25
Iteration 1: EE criterion = 2.903e-25 (backed up)
Endogenous treatment-effects estimation          Number of obs = 935
Outcome model: exponential
Treatment model: probit
```

		Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE							
	urban (1 vs 0)	481.0465	31.74882	15.15	0.000	418.82	543.2731
P0mean							
	urban 0	233.8083	13.51028	17.31	0.000	207.3286	260.288

When everyone lives outside urban areas, wages are \$234 a month on average. Wages are \$481 a month greater, on average, when everyone lives in urban areas.



## Stored results

eteffects stores the following in e():

### Scalars

e(N) number of observations  
 e(nj) number of observations for treatment level *j*  
 e(k\_eq) number of equations in e(b)  
 e(k\_levels) number of levels in treatment variable  
 e(rank) rank of e(V)  
 e(converged) 1 if converged, 0 otherwise

### Macros

e(cmd) eteffects  
 e(cmdline) command as typed  
 e(depvar) name of outcome variable  
 e(tvar) name of treatment variable  
 e(omodel) fractional, linear, probit, or exponential  
 e(stat) statistic estimated, ate, atet, or pomeans  
 e(wtype) weight type  
 e(wexp) weight expression  
 e(title) title in estimation output  
 e(tlevels) levels of treatment variable  
 e(vce) *vctype* specified in vce()  
 e(vcetype) title used to label Std. err.  
 e(properties) b V  
 e(estat\_cmd) program used to implement estat  
 e(predict) program used to implement predict  
 e(marginsnotok) predictions disallowed by margins  
 e(asbalanced) factor variables fvset as asbalanced  
 e(asobserved) factor variables fvset as asobserved

### Matrices

e(b) coefficient vector  
 e(V) variance–covariance matrix of the estimators

Functions  
`e(sample)` marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices  
`r(table)` matrix containing the coefficients with their standard errors, test statistics,  $p$ -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r`-class command is run after the estimation command.

## Methods and formulas

The treatment-effects models considered in `eteffects` are given by

$$y_{i0} = E(y_{i0}|\mathbf{x}_i) + \epsilon_{i0} \quad (1)$$

$$y_{i1} = E(y_{i1}|\mathbf{x}_i) + \epsilon_{i1} \quad (2)$$

$$t_i = E(t_i|\mathbf{z}_i) + \nu_i \quad (3)$$

$$y_i = t_i y_{i1} + (1 - t_i) y_{i0} \quad (4)$$

$$E(\epsilon_{ij}|\mathbf{x}_i, \mathbf{z}_i) = E(\epsilon_{ij}|\mathbf{z}_i) = E(\epsilon_{ij}|\mathbf{x}_i) = 0 \quad \text{for } j \in \{0, 1\} \quad (5)$$

$$E(\epsilon_{ij}|t) \neq 0 \quad \text{for } j \in \{0, 1\} \quad (6)$$

where the subscript  $i$  denotes individual level observations,  $y_{i1}$  is the potential outcome of receiving the treatment,  $y_{i0}$  is the potential outcome when the treatment is not received,  $t_i$  is the observed binary treatment, and  $y_i$  is the observed outcome. Each one of the potential outcomes is determined by its expected value conditional on a set of regressors  $\mathbf{x}_i$  and an unobserved random component  $\epsilon_{ij}$ , for  $j \in \{0, 1\}$ . Similarly, the treatment is given by its expectation conditional on a set of regressors  $\mathbf{z}_i$ , which does not need to differ from  $\mathbf{x}_i$ , and an unobserved component  $\nu_i$ .

Equations (1)–(5) describe the parametric treatment-effects models in `[TE] teffects`. Equation (6) adds endogeneity to the framework. It states that the unobservables in the potential-outcome equations are correlated to treatment status. For our birthweight example, this would happen if mothers who do not smoke are more health conscious than those who smoke and if we do not observe health awareness in our data. If we do not observe health awareness, the decision to smoke or not to smoke is not independent of the infant's birthweight.

Equations (3), (5), and (6) are the basis of the control-function estimator implemented by `eteffects`. Equation (5) states that the unobserved components in the potential outcome are independent of  $\mathbf{z}_i$ . Therefore, the correlation between  $t_i$  and the unobserved components must be equivalent to the correlation between  $\epsilon_{ij}$  and  $\nu_i$ . Another way of stating this is

$$\text{from (3)} \quad E(\epsilon_{ij}|t_i) = E(\epsilon_{ij}|E(t|\mathbf{z}_i) + \nu_i)$$

$$\text{from (5)} \quad = E(\epsilon_{ij}|\nu_i)$$

$$= \nu_i \beta_{2j}$$

We fit (3) using a probit estimator. We then obtain  $\hat{\nu}_i$  as the difference between the treatment and our estimate of  $E(t_i|\mathbf{z}_i)$  and use this statistic to compute an estimate of  $E(y_{ij}|\mathbf{x}_i, \nu_i, t_i)$  for  $j \in \{0, 1\}$ . If the outcome is linear, for instance,

$$E(y_{ij}|\mathbf{x}_i, \nu_i, t_i = j) = \mathbf{x}'_i \beta_{1j} + \nu_i \beta_{2j} \quad \text{for } j \in \{0, 1\} \quad (7)$$



For the probit and exponential-mean cases, respectively, we have the following:

$$E(y_{ij}|\mathbf{x}_i, \nu_i, t_i = j) = \Phi(\mathbf{x}'_i\beta_{1j} + \nu_i\beta_{2j}) \quad (8)$$

$$E(y_{ij}|\mathbf{x}_i, \nu_i, t_i = j) = \exp(\mathbf{x}'_i\beta_{1j} + \nu_i\beta_{2j}) \quad (9)$$

The parameters of (3) and (7)–(9), and the ATE, ATET, and POMs are estimated using the generalized method of moments (GMM). The moment equations used in GMM are the sample analogs of  $E\{\mathbf{w}'_i\epsilon_i(\theta)\} = 0$ , where  $\mathbf{w}_i$  are the instruments,  $\epsilon_i(\theta)$  are residuals, and  $\theta$  are the parameters of the model (see [R] **gmm**). The moment conditions in the GMM estimation for the linear model are given by

$$\frac{1}{n} \sum_{i=1}^n \mathbf{x}'_i (y_i - \mathbf{x}'_i\widehat{\beta}_{1j} + \widehat{\nu}_i\widehat{\beta}_{2j})t_i = 0 \quad (10)$$

$$\frac{1}{n} \sum_{i=1}^n \mathbf{x}'_i (y_i - \mathbf{x}'_i\widehat{\beta}_{1j} + \widehat{\nu}_i\widehat{\beta}_{2j})(1 - t_i) = 0 \quad (11)$$

$$\frac{1}{n} \sum_{i=1}^n \mathbf{z}'_i \left\{ t_i \frac{\phi(\mathbf{z}'_i\widehat{\pi})}{\Phi(\mathbf{z}'_i\widehat{\pi})} - (1 - t_i) \frac{\phi(\mathbf{z}'_i\widehat{\pi})}{1 - \Phi(\mathbf{z}'_i\widehat{\pi})} \right\} = 0 \quad (12)$$

$$\frac{1}{n} \sum_{i=1}^n \left\{ (\mathbf{x}'_i\widehat{\beta}_{10} + \widehat{\nu}_i\widehat{\beta}_{20}) - \widehat{\text{POM0}} \right\} = 0 \quad (13)$$

$$\frac{1}{n} \sum_{i=1}^n \left\{ (\mathbf{x}'_i\widehat{\beta}_{11} + \widehat{\nu}_i\widehat{\beta}_{21}) - \widehat{\text{POM0}} - \widehat{\text{ATE}} \right\} = 0 \quad (14)$$

where  $\widehat{\nu}_i = t_i - \Phi(\mathbf{z}'_i\widehat{\pi})$ ,  $n$  is the number of observations, and  $\widehat{\beta}_{11}, \widehat{\beta}_{10}, \widehat{\beta}_{21}, \widehat{\beta}_{20}, \widehat{\pi}, \widehat{\text{ATE}}$ , and  $\widehat{\text{POM0}}$  are the parameters. If we want to estimate the ATET, we replace (14) with

$$\frac{1}{n} \sum_{i=1}^n \left\{ (\mathbf{x}'_i\widehat{\beta}_{11} + \widehat{\nu}_i\widehat{\beta}_{21}) \frac{n}{n_t} - \widehat{\text{POM0}} \frac{n}{n_t} - \widehat{\text{ATE}} \right\} = 0 \quad (15)$$

and if we want to estimate the potential-outcome means, we replace (14) with

$$\frac{1}{n} \sum_{i=1}^n \left\{ (\mathbf{x}'_i\widehat{\beta}_{11} + \widehat{\nu}_i\widehat{\beta}_{21}) - \widehat{\text{POM1}} \right\} = 0 \quad (16)$$

where  $\widehat{\text{ATE}}$  and  $\widehat{\text{POM1}}$  are the parameters of the model, and  $n_t$  is the number of treated units.

For the exponential-mean outcome model, we replace  $\mathbf{x}'_i\widehat{\beta}_{1j} + \widehat{\nu}_i\widehat{\beta}_{2j}$  with  $\exp(\mathbf{x}'_i\widehat{\beta}_{1j} + \widehat{\nu}_i\widehat{\beta}_{2j})$  to obtain the residual equations in (10)–(16). For the probit outcome model, we replace (10) and (11) with the following:

$$\frac{1}{n} \sum_{i=1}^n t_i \mathbf{x}'_i \left\{ y_i \frac{\phi(\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j})}{\Phi(\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j})} - (1 - y_i) \frac{\phi(\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j})}{1 - \Phi(\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j})} \right\} = 0$$

$$\frac{1}{n} \sum_{i=1}^n (1 - t_i) \mathbf{x}'_i \left\{ y_i \frac{\phi(\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j})}{\Phi(\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j})} - (1 - y_i) \frac{\phi(\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j})}{1 - \Phi(\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j})} \right\} = 0$$

For the remaining equations,  $\mathbf{x}'_i\widehat{\beta}_{1j} + \widehat{\nu}_i\widehat{\beta}_{2j}$  is replaced with  $\Phi(\mathbf{x}'_i\widehat{\beta}_{1j} + \widehat{\nu}_i\widehat{\beta}_{2j})$ . The fractional probit model uses the same moment conditions as the probit model.

## Acknowledgment

We thank Jeffrey Wooldridge of Michigan State University for helpful conversations and advice about the literature and methods in this area.

## References

- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cerulli, G. 2014. `ivtreatreg`: A command for fitting binary treatment models with heterogeneous response to treatment and unobservable selection. *Stata Journal* 14: 453–480.
- . 2015. *Econometric Evaluation of Socio-Economic Programs: Theory and Applications*. Berlin: Springer.
- Drukker, D. M. 2016. A generalized regression-adjustment estimator for average treatment effects from panel data. *Stata Journal* 16: 826–836.
- Terza, J. V., A. Basu, and P. J. Rathouz. 2008. Two-stage residual inclusion estimation: Addressing endogeneity in health econometric modeling. *Journal of Health Economics* 27: 531–543. <https://doi.org/10.1016/j.jhealeco.2007.09.009>.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.
- . 2014. Quasi-maximum likelihood estimation and testing for nonlinear models with endogenous explanatory variables. *Journal of Econometrics* 182: 226–234. <https://doi.org/10.1016/j.jeconom.2014.04.020>.

## Also see

- [TE] **eteffects postestimation** — Postestimation tools for eteffects
- [TE] **etregress** — Linear regression with endogenous treatment effects
- [TE] **teffects** — Treatment-effects estimation for observational data
- [R] **gmm** — Generalized method of moments estimation
- [R] **probit** — Probit regression
- [R] **regress** — Linear regression
- [U] **20 Estimation and postestimation commands**

[Postestimation commands](#)[predict](#)[estat](#)[Remarks and examples](#)[Also see](#)

## Postestimation commands

The following postestimation command is of special interest after `eteffects`:

Command	Description
<code>estat endogenous</code>	perform tests of endogeneity

The following standard postestimation commands are available after `eteffects`:

Command	Description
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
<code>hausman</code>	Hausman’s specification test
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of coefficients
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
<code>predict</code>	treatment effects, conditional means at treatment, propensity scores, etc.
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

# predict

## Description for predict

`predict` creates a new variable containing predictions such as treatment effects, conditional means, propensity scores, and linear predictions.

## Menu for predict

Statistics > Postestimation

## Syntax for predict

```
predict [type] { stub* | newvar | newvarlist } [if] [in] [, statistic tlevel]
```

```
predict [type] stub* [if] [in], scores
```

<i>statistic</i>	Description
------------------	-------------

Main

<code>te</code>	treatment effect; the default
<code>cmean</code>	conditional mean at treatment level
<code>ps</code>	propensity score
<code>xb</code>	linear prediction
<code>psxb</code>	linear prediction for propensity score
<code>xbtotal</code>	linear prediction, using residuals from treatment model

Specify one new variable with `te`; specify one or two new variables with `cmean`, `ps`, and `xb`.

## Options for predict

Main

`te`, the default, calculates the treatment effect.

`cmean` calculates the conditional mean for the control group. To also obtain the conditional mean for the treatment group, specify two variables. If you want the conditional mean for only the treatment group, specify the `tlevel` option.

`ps` calculates the probability of being in the control group. To also obtain the probability of being in the treatment group, specify two variables. If you want the probability of being in the treatment group only, specify the `tlevel` option.

`xb` calculates the linear prediction for the control group. To also obtain the linear prediction for the treatment group, specify two variables. If you want the linear prediction for only the treatment group, specify the `tlevel` option.

`psxb` calculates the linear prediction for the propensity score.

`xbtotal` calculates the linear prediction for the control group, including the residuals from the treatment model as regressors. To also obtain the linear prediction for the treatment group, specify two variables. If you want the linear prediction, including the residuals from the treatment model as regressors, only for the treatment group, specify the `tlevel` option.

`tlevel` specifies that the statistic be calculated for the treatment group; the default is to calculate the statistic for the control group.

`scores` calculates the score variables. For `eteffects`, this is the same as the residuals in the moment conditions used by the generalized method of moments (see [R] [gmm](#)). For the average treatment effect, the average treatment effect on the treated, and the potential-outcome means, parameter-level scores are computed. For the auxiliary equations, equation-level scores are computed.

## estat

### Description for estat

`estat endogenous` performs a Wald test to determine whether the estimated correlations between the treatment-assignment and potential-outcome models are different from zero. The null hypothesis is that the correlations are jointly zero. Rejection of the null hypothesis suggests endogeneity.

### Menu for estat

Statistics > Postestimation

### Syntax for estat

```
estat endogenous
```

`collect` is allowed with `estat endogenous`; see [U] [11.1.10 Prefix commands](#).

## Remarks and examples

### ▶ Example 1: Testing for endogeneity

In [example 3](#) of [TE] [eteffects](#), endogeneity could arise if unobservable factors that determine wages are correlated with the decision to live in an urban area. If there is no endogeneity, we would prefer to use one of the `teffects` estimators because they will give us the more efficient standard errors. The control-function approach used by `eteffects` allows us to test for endogeneity.

The control-function approach estimates the correlation between the unobservables of the treatment-assignment and potential-outcome models. If there is no correlation between the unobservables, then there is no endogeneity. We test for correlation, and thus for endogeneity, by typing

```
. use https://www.stata-press.com/data/r17/nlsy80
. eteffects (wage exper iq i.college, exponential nocons)
> (urban i.college fcollege)
(output omitted)
. estat endogenous

Test of endogeneity
H0: Treatment and outcome unobservables are uncorrelated

      chi2( 2) = 275.36
      Prob > chi2 = 0.0000
```

We reject the null hypothesis of no endogeneity. This suggests that unobservable factors that determine wages mediate the decision to live in an urban area.

◀

## □ Technical note

The estimated correlations between the unobservables of the treatment-assignment and potential-outcome models are auxiliary parameters. They appear under the headings TE0M0 and TE0M1, which refer to treatment residuals (TE) for outcome model 0 (OM0) and outcome model 1 (OM1), when the option `aequations` is specified.

For the model in [example 3](#) of [\[TE\] eteffects](#) with the `aequations` option, the results are the following:

```
. eteffects (wage exper iq i.college, exponential nocons)
> (urban i.college fcollege), aequations
Iteration 0: EE criterion = 2.903e-25
Iteration 1: EE criterion = 2.903e-25 (backed up)
Endogenous treatment-effects estimation      Number of obs = 935
Outcome model: exponential
Treatment model: probit
```

wage		Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE							
urban		481.0465	31.74882	15.15	0.000	418.82	543.2731
(1 vs 0)							
POmean							
urban		233.8083	13.51028	17.31	0.000	207.3286	260.288
0							
TME1							
college		.195811	.1012119	1.93	0.053	-.0025607	.3941827
1							
fcollege		.1069748	.0992075	1.08	0.281	-.0874683	.3014179
_cons		.498012	.056408	8.83	0.000	.3874543	.6085698
OME0							
exper		.0193244	.0085633	2.26	0.024	.0025405	.0361082
iq		.0099473	.0036949	2.69	0.007	.0027053	.0171892
college							
1		-.3718598	.2678636	-1.39	0.165	-.8968629	.1531433
OME1							
exper		.0238566	.017597	1.36	0.175	-.0106329	.058346
iq		.0148581	.0113311	1.31	0.190	-.0073505	.0370667
college							
1		1.236947	.6401383	1.93	0.053	-.0177013	2.491595
TE0M0							
_cons		-7.771932	.6406251	-12.13	0.000	-9.027534	-6.51633
TE0M1							
_cons		16.7739	4.777519	3.51	0.000	7.410131	26.13766

Among other things, we can use these correlations to test the joint significance of the coefficients on the residuals from the treatment-assignment models. This is equivalent to the endogeneity test in [example 1](#). We type

```
. test [TEOM0]_cons [TEOM1]_cons
( 1)  [TEOM0]_cons = 0
( 2)  [TEOM1]_cons = 0
      chi2( 2) = 275.36
      Prob > chi2 = 0.0000
```

□

## Also see

[\[TE\] eteffects](#) — Endogenous treatment-effects estimation

[\[U\] 20 Estimation and postestimation commands](#)

# Title

**etpoisson** — Poisson regression with endogenous treatment effects

[Description](#)  
[Options](#)  
[References](#)

[Quick start](#)  
[Remarks and examples](#)  
[Also see](#)

[Menu](#)  
[Stored results](#)

[Syntax](#)  
[Methods and formulas](#)

## Description

`etpoisson` estimates the parameters of a Poisson regression model in which one of the regressors is an endogenous binary treatment. Both the average treatment effect and the average treatment effect on the treated can be estimated with `etpoisson`.

## Quick start

Poisson model of  $y$  on  $x$  and endogenous binary treatment `treat` modeled by  $x$  and  $w$

```
etpoisson y x, treat(treat = x w)
```

With robust standard errors

```
etpoisson y x, treat(treat = x w) vce(robust)
```

Average treatment effect after `etpoisson` with the required `vce(robust)` option

```
margins r.treat, vce(unconditional)
```

As above, but calculate average treatment effect on the treated

```
margins, vce(unconditional) predict(cte) subpop(if treat==1)
```

## Menu

Statistics > Treatment effects > Endogenous treatment > Maximum likelihood estimator > Count outcomes



## Syntax

```
etpoisson devar [indepvars] [if] [in] [weight],
      treat(devart = indepvarst [, noconstant offset(varnameo)] ) [options]
```

<i>options</i>	Description
Model	
* <u>treat</u> ( )	equation for treatment effects
<u>noconstant</u>	suppress constant term
<u>exposure</u> ( <i>varname</i> <sub><i>e</i></sub> )	include ln( <i>varname</i> <sub><i>e</i></sub> ) in model with coefficient constrained to 1
<u>offset</u> ( <i>varname</i> <sub><i>o</i></sub> )	include <i>varname</i> <sub><i>o</i></sub> in model with coefficient constrained to 1
<u>constraints</u> ( <i>constraints</i> )	apply specified linear constraints
SE/Robust	
<u>vce</u> ( <i>vcetype</i> )	<i>vcetype</i> may be oim, <u>robust</u> , <u>cluster</u> <i>clustvar</i> , opg, <u>bootstrap</u> , or <u>jackknife</u>
Reporting	
<u>level</u> (#)	set confidence level; default is level(95)
<u>irr</u>	report incidence-rate ratios
<u>nocnsreport</u>	do not display constraints
<u>display_options</u>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
<u>intpoints</u> (#)	use # Gauss–Hermite quadrature points; default is intpoints(24)
Maximization	
<u>maximize_options</u>	control the maximization process; seldom used
<u>collinear</u>	keep collinear variables
<u>coeflegend</u>	display legend instead of statistics

\*treat( ) is required.

The full specification is treat(*devar*<sub>*t*</sub> = *indepvars*<sub>*t*</sub> [, noconstant offset(*varname*<sub>*o*</sub>)] ).

*indepvars* and *indepvars*<sub>*t*</sub> may contain factor variables; see [U] 11.4.3 Factor variables.

*devar*, *devar*<sub>*t*</sub>, *indepvars*, and *indepvars*<sub>*t*</sub> may contain time-series operators; see [U] 11.4.4 Time-series varlists.

bootstrap, by, collect, jackknife, rolling, statsby, and svy are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the bootstrap prefix; see [R] bootstrap.

aweights are not allowed with the jackknife prefix; see [R] jackknife.

vce( ) and weights are not allowed with the svy prefix; see [SVY] svy.

fweights, aweights, iweights, and pweights are allowed; see [U] 11.1.6 weight.

collinear and coeflegend do not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

## Options

### Model

`treat(depvart = indepvarst [, noconstant offset(varnameo)])` specifies the variables and options for the treatment equation. It is an integral part of specifying a treatment-effects model and is required.

The indicator of treatment, *depvar*<sub>*t*</sub>, should be coded as 0 or 1.

*noconstant*, *exposure*(*varname*<sub>*e*</sub>), *offset*(*varname*<sub>*o*</sub>), *constraints*(*constraints*); see [R] **Estimation options**.

### SE/Robust

*vce*(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (*oim*, *opg*), that are robust to some kinds of misspecification (*robust*), that allow for intragroup correlation (*cluster* *clustvar*), and that use bootstrap or jackknife methods (*bootstrap*, *jackknife*); see [R] *vce\_option*.

### Reporting

*level*(#); see [R] **Estimation options**.

*irr* reports estimated coefficients transformed to incidence-rate ratios, that is,  $e^{\beta_i}$  rather than  $\beta_i$ . Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated or stored. *irr* may be specified at estimation or when replaying previously estimated results.

*nocnsreport*; see [R] **Estimation options**.

*display\_options*: *noci*, *nopvalues*, *noomitted*, *vsquish*, *noemptycells*, *baselevels*, *allbaselevels*, *nofvlabel*, *fvwrap*(#), *fvwrapon*(*style*), *cformat*(*%fmt*), *pformat*(*%fmt*), *sformat*(*%fmt*), and *nolstretch*; see [R] **Estimation options**.

### Integration

*intpoints*(#) specifies the number of integration points to use for integration by quadrature. The default is *intpoints*(24); the maximum is *intpoints*(128). Increasing this value improves the accuracy but also increases computation time. Computation time is roughly proportional to its value.

### Maximization

*maximize\_options*: *difficult*, *technique*(*algorithm\_spec*), *iterate*(#), [*no*] *log*, *trace*, *gradient*, *showstep*, *hessian*, *showtolerance*, *tolerance*(#), *ltolerance*(#), *nrtolerance*(#), *nonrtolerance*, and *from*(*init\_specs*); see [R] **Maximize**. These options are seldom used.

Setting the optimization type to *technique*(*bhhh*) resets the default *vcetype* to *vce*(*opg*).

The following options are available with *etpoisson* but are not shown in the dialog box:

*collinear*, *coeflegend*; see [R] **Estimation options**.

## Remarks and examples

Remarks are presented under the following headings:

[Overview](#)

[Basic example](#)

[Average treatment effect \(ATE\)](#)

[Average treatment effect on the treated \(ATET\)](#)

### Overview

`etpoisson` estimates the parameters of a Poisson regression model that includes an endogenous binary-treatment variable. The dependent variable must be a Poisson distributed count. The parameters estimated by `etpoisson` can be used to estimate the average treatment effect (ATE) and average treatment effect on the treated (ATET).

We call the model fit by `etpoisson` an endogenous treatment-regression model, although it is also known as an endogenous binary-variable model or as an endogenous dummy-variable model. The endogenous treatment-regression model fit by `etpoisson` is a specific endogenous treatment-effects model; it uses a nonlinear model for the outcome and a constrained normal distribution to model the deviation from the conditional independence assumption imposed by the estimators implemented by `teffects`; see [TE] [teffects intro](#). In treatment-effects jargon, the endogenous binary-variable model fit by `etpoisson` is a nonlinear potential-outcome model that allows for a specific correlation structure between the unobservables that affect the treatment and the unobservables that affect the potential outcomes. See [TE] [etregress](#) for an estimator that allows for a linear-outcome model and a similar model for the endogeneity of the treatment.

More formally, we have an equation for outcome  $y_j$  and an equation for treatment  $t_j$ :

$$E(y_j | \mathbf{x}_j, t_j, \epsilon_j) = \exp(\mathbf{x}_j \boldsymbol{\beta} + \delta t_j + \epsilon_j)$$

$$t_j = \begin{cases} 1, & \mathbf{w}_j \boldsymbol{\gamma} + u_j > 0 \\ 0, & \text{otherwise} \end{cases}$$

The  $\mathbf{x}_j$  are the covariates used to model the outcome,  $\mathbf{w}_j$  are the covariates used to model treatment assignment, and error terms  $\epsilon_j$  and  $u_j$  are bivariate normal with mean 0 and covariance matrix

$$\begin{bmatrix} \sigma^2 & \sigma\rho \\ \sigma\rho & 1 \end{bmatrix}$$

The covariates  $\mathbf{x}_j$  and  $\mathbf{w}_j$  are unrelated to the error terms; in other words, they are exogenous. Note that  $y_j$  may be a count or continuous and nonnegative in this specification.

[Terza \(1998\)](#) describes the maximum likelihood estimator used in `etpoisson`. [Terza \(1998\)](#) categorized the model fit by `etpoisson` as an endogenous-switching model. These models involve a binary switch that is endogenous for the outcome. Calculation of the maximum likelihood estimate involves numeric approximation of integrals via Gauss–Hermite quadrature. This is computationally intensive, but the computational costs are reasonable on modern computers.

## Basic example

### ▷ Example 1

In this example, we observe a simulated random sample of 5,000 households. The outcome of interest is the number of trips taken by members of the household in the 24-hour period immediately prior to the interview time.

We have fictional household level data on the following variables: number of trips taken in the past 24 hours (`trips`), distance to the central business district from the household (`cbd`), distance from the household to a public transit node (`ptn`), an indicator of whether there is a full-time worker in the household (`worker`), an indicator of whether the examined period is on a weekend (`weekend`), the ratio of the household income to the median income of the census tract (`realinc`), and an indicator of car ownership (`owncar`). We suspect that unobservables that affect the number of trips also affect the household's propensity to own a car.

We use `etpoisson` to estimate the parameters of a Poisson regression model for the number of trips with car ownership as an endogenous treatment. In subsequent examples, we will use `margins` (see [R] `margins`) to estimate the ATE and the ATET of car ownership on the number of trips taken by the household. In the `etpoisson` command below, we specify the `vce(robust)` option because we need to specify `vce(unconditional)` when we use `margins` later.

```
. use https://www.stata-press.com/data/r17/trip1
(Household trips, car ownership)
. etpoisson trips cbd ptn worker weekend,
> treat(owncar = cbd ptn worker realinc) vce(robust)
Iteration 0: log pseudolikelihood = -14845.147 (not concave)
Iteration 1: log pseudolikelihood = -14562.997 (not concave)
Iteration 2: log pseudolikelihood = -13655.592 (not concave)
Iteration 3: log pseudolikelihood = -12847.219 (not concave)
Iteration 4: log pseudolikelihood = -12566.037
Iteration 5: log pseudolikelihood = -12440.974
Iteration 6: log pseudolikelihood = -12413.485
Iteration 7: log pseudolikelihood = -12412.699
Iteration 8: log pseudolikelihood = -12412.696
Iteration 9: log pseudolikelihood = -12412.696
Poisson regression with endogenous treatment          Number of obs = 5,000
(24 quadrature points)                               Wald chi2(5) = 397.94
Log pseudolikelihood = -12412.696                   Prob > chi2 = 0.0000
```

	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
<b>trips</b>						
cbd	-.0100919	.0020071	-5.03	0.000	-.0140258	-.006158
ptn	-.0204038	.0020289	-10.06	0.000	-.0243805	-.0164272
worker	.692301	.0548559	12.62	0.000	.5847854	.7998166
weekend	.0930517	.034538	2.69	0.007	.0253585	.160745
1.owncar	.5264713	.1124157	4.68	0.000	.3061406	.746802
_cons	-.2340772	.0810812	-2.89	0.004	-.3929934	-.0751609
<b>owncar</b>						
cbd	.007218	.00239	3.02	0.003	.0025337	.0119023
ptn	.0084769	.0024518	3.46	0.001	.0036714	.0132824
worker	.543643	.0504267	10.78	0.000	.4448085	.6424774
realinc	.176479	.0108746	16.23	0.000	.1551652	.1977928
_cons	-.4611246	.0592161	-7.79	0.000	-.5771859	-.3450633
/athrho	.5741169	.0957832	5.99	0.000	.3863852	.7618486
/lnsigma	-.2182037	.0256281	-8.51	0.000	-.2684338	-.1679735
rho	.5183763	.0700449			.3682398	.6421645
sigma	.8039617	.020604			.764576	.8453762

Wald test of indep. eqns. (rho = 0): chi2(1) = 35.93 Prob > chi2 = 0.0000

The Wald test in the header is highly significant, indicating a good model fit. All the covariates are statistically significant, and the Wald test in the footer indicates that we can reject the null hypothesis of no correlation between the treatment errors and the outcome errors.

We can interpret the coefficient on 1.owncar as the logarithm of the ratio of the treatment potential-outcome mean to the control potential-outcome mean. The treatment variable did not interact with any of the outcome covariates, so the effect of each regressor is the same in the two regimes and will cancel from the ratio of potential-outcome means. This means the ratio is equivalent to the exponentiated coefficient on 1.owncar. After discussing the other parameters, we will use `lincom` to obtain this ratio. See [R] [lincom](#) for more information.

The estimated correlation between the treatment-assignment errors and the outcome errors is 0.518, indicating that unobservables that increase the number of trips tend to occur with unobservables that increase the chance of car ownership.

The results for the two ancillary parameters require explanation. `etpoisson` estimates the inverse hyperbolic tangent of  $\rho$ ,

$$\operatorname{atanh} \rho = \frac{1}{2} \ln \left( \frac{1 + \rho}{1 - \rho} \right)$$

and  $\ln \sigma$  rather than  $\rho$  and  $\sigma$ . For numerical stability during optimization, `etpoisson` does not directly estimate  $\rho$  and  $\sigma$ .

Now we use `lincom` and the `eform` option to estimate the exponentiated coefficient for `1.owncar`. This corresponds to the ratio of the treatment regime potential-outcome mean to the control regime potential-outcome mean.

```
. lincom [trips]_b[1.owncar], eform
(1) [trips]1.owncar = 0
```

	exp(b)	Std. err.	z	P> z	[95% conf. interval]	
(1)	1.692948	.1903139	4.68	0.000	1.358173	2.110241

The potential-outcome mean for the treatment regime is 1.69 times the potential-outcome mean for the control regime. So the average number of trips in the treatment regime is over one and a half times the average number of trips in the control regime.

By interacting the treatment, `owncar`, with the other regressors, we could estimate different coefficients for the regressors in the treatment and control regimes. In the current model, there are no treatment interactions, so the coefficients are the same in each regime.

◀

## Average treatment effect (ATE)

The parameter estimates from `etpoisson` can be used by `margins` to estimate the ATE, the average difference of the treatment and control potential outcomes.

### ► Example 2

Continuing with [example 1](#), we use `margins` to estimate the ATE of car ownership on the number of trips taken in a 24-hour period.

We can estimate the ATE of car ownership by using the potential-outcome means obtained through the `predict`, `pomean` command and the `margins` command; see [Methods and formulas](#) below and [\[TE\] etpoisson postestimation](#) for more details about the use of `predict` after `etpoisson`.

The `r.` notation indicates that the potential-outcome means for treatment and control will be contrasted. We specify the `contrast(nowald)` option to suppress the Wald tests that `margins` displays by default for contrasts.

```
. margins r.owncar, vce(unconditional) contrast(nowald)
Contrasts of predictive margins                               Number of obs = 5,000
Expression: Potential-outcome mean, predict()
```

	Unconditional			
	Contrast	std. err.	z	[95% conf. interval]
owncar (1 vs 0)	1.058914	.1922909	.6820309	1.435797

The estimated ATE of car ownership on the number of trips taken is 1.06. The average household will take 1.06 more trips when it owns a car.

◀

## Average treatment effect on the treated (ATET)

The parameter estimates from `etpoisson` can be used by `margins` to estimate the ATET, the average difference of the treatment and control potential outcomes in the treated population.

### ▷ Example 3

Continuing with the [previous example](#), we use `margins` to estimate the ATET of car ownership on the number of trips taken in a 24-hour period.

We can estimate the ATET of car ownership by using the conditional treatment effect (conditional on exogenous covariates and treatment level) obtained through the `predict`, `cte` command and the `margins` command; see [Methods and formulas](#) below and [TE] [etpoisson postestimation](#) for more details about the use of `predict` after `etpoisson`.

We estimate the ATET with `margins`. We specify `cte` in the `predict()` option. Estimation is restricted to the treated subpopulation by specifying `owncar` in the `subpop()` option.

```
. margins, predict(cte) vce(unconditional) subpop(owncar)
Predictive margins                                Number of obs = 5,000
                                                Subpop. no. obs = 3,504
Expression: Conditional treatment effect, predict(cte)
```

	Unconditional				[95% conf. interval]	
	Margin	std. err.	z	P> z		
_cons	1.251971	.2059201	6.08	0.000	.8483747	1.655567

The estimated ATET of car ownership on the number of trips taken is 1.25. Thus the average household in the treated population will take 1.25 more trips than it would if it did not own a car. This number is higher than the ATE. In this model, the ATE and ATET will only coincide when there is no correlation between the treatment errors and outcome errors and the exogenous covariates  $x$  have the same distribution in the general population and treated subpopulation. See [Methods and formulas](#) for more details.

◀

## Stored results

`etpoisson` stores the following in `e()`:

### Scalars

<code>e(N)</code>	number of observations
<code>e(k)</code>	number of parameters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_eq_model)</code>	number of equations in overall model test
<code>e(k_aux)</code>	number of auxiliary parameters
<code>e(k_dv)</code>	number of dependent variables
<code>e(df_m)</code>	model degrees of freedom
<code>e(ll)</code>	log likelihood
<code>e(N_clust)</code>	number of clusters
<code>e(chi2)</code>	$\chi^2$
<code>e(chi2_c)</code>	$\chi^2$ for comparison, $\rho=0$ test
<code>e(n_quad)</code>	number of quadrature points
<code>e(p)</code>	$p$ -value for model test
<code>e(p_c)</code>	$p$ -value for comparison test
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(ic)</code>	number of iterations
<code>e(rc)</code>	return code
<code>e(converged)</code>	1 if converged, 0 otherwise

### Macros

<code>e(cmd)</code>	<code>etpoisson</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(title2)</code>	secondary title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(offset1)</code>	offset for regression equation
<code>e(offset2)</code>	offset for treatment equation
<code>e(chi2type)</code>	Wald; type of model $\chi^2$ test
<code>e(chi2_ct)</code>	Wald; type of comparison $\chi^2$ test
<code>e(vce)</code>	<i>vce</i> type specified in <code>vce()</code>
<code>e(vctype)</code>	title used to label Std. err.
<code>e(opt)</code>	type of optimization
<code>e(which)</code>	max or min; whether optimizer is to perform maximization or minimization
<code>e(ml_method)</code>	type of ml method
<code>e(user)</code>	name of likelihood-evaluator program
<code>e(technique)</code>	maximization technique
<code>e(properties)</code>	<code>b V</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsok)</code>	predictions allowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

### Matrices

<code>e(b)</code>	coefficient vector
<code>e(Cns)</code>	constraints matrix
<code>e(ilog)</code>	iteration log (up to 20 iterations)
<code>e(gradient)</code>	gradient vector
<code>e(V)</code>	variance-covariance matrix of the estimators
<code>e(V_modelbased)</code>	model-based variance

### Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------



In addition to the above, the following is stored in `r()`:

Matrices	
<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, $p$ -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

## Methods and formulas

Terza (1998) derives the maximum likelihood estimator implemented here. We provide some details of the derivation and then explain how the model is nested in a more general potential-outcomes model. Then the ATE and ATET are derived.

Let  $\mathbf{x}_j$  be the covariates used to model the outcome, and let  $\mathbf{w}_j$  be the covariates used to model treatment assignment. Define  $\mathbf{z}_j = (\mathbf{w}_j, \mathbf{x}_j)$ . The vector  $\mathbf{z}_j$  contains all the exogenous covariates in the model. When offsets  $o_j^\beta$  are used in the outcome variable equation, the following formulas apply with  $\mathbf{x}_j\beta$  changed to  $\mathbf{x}_j\beta + o_j^\beta$ . Similarly, when offsets  $o_j^\gamma$  are used in the endogenous treatment equation, the following formulas apply with  $\mathbf{w}_j\gamma$  changed to  $\mathbf{w}_j\gamma + o_j^\gamma$ . If offsets are used in either equation, they are included in the vector of exogenous covariates  $\mathbf{z}_j$ .

For treatment  $t_j$ ,  $\mathbf{z}_j$ , and  $\epsilon_j$ , outcome  $y_j$  of this model has conditional mean

$$E(y_j | \mathbf{z}_j, t_j, \epsilon_j) = \exp(\mathbf{x}_j\beta + \delta t_j + \epsilon_j) \tag{1}$$

The probability density function of  $y_j$  for this model, conditioned on treatment  $t_j$ ,  $\mathbf{z}_j$ , and  $\epsilon_j$ , is given by

$$f(y_j | \mathbf{z}_j, t_j, \epsilon_j) = \frac{\exp\{-\exp(\mathbf{x}_j\beta + \delta t_j + \epsilon_j)\} \{\exp(\mathbf{x}_j\beta + \delta t_j + \epsilon_j)\}^{y_j}}{y_j!}$$

The treatment  $t_j$  is determined by

$$t_j = \begin{cases} 1, & \text{if } \mathbf{w}_j\gamma + u_j > 0 \\ 0, & \text{otherwise} \end{cases}$$

The error terms  $\epsilon_j$  and  $u_j$  are bivariate normal with mean zero and covariance matrix

$$\begin{bmatrix} \sigma^2 & \sigma\rho \\ \sigma\rho & 1 \end{bmatrix}$$

Conditional on  $\epsilon_j$ ,  $u_j$  is normal with mean  $\epsilon_j\rho/\sigma$  and variance  $(1 - \rho^2)$ ; thus we obtain the following conditional probability density for  $t_j$ :

$$\Pr(t_j | \mathbf{z}_j, \epsilon_j) = t_j \Phi \left\{ \frac{\mathbf{w}_j\gamma + (\rho/\sigma)\epsilon_j}{\sqrt{1 - \rho^2}} \right\} + (1 - t_j) \left[ 1 - \Phi \left\{ \frac{\mathbf{w}_j\gamma + (\rho/\sigma)\epsilon_j}{\sqrt{1 - \rho^2}} \right\} \right]$$

$\Phi$  denotes the standard normal cumulative distribution function. This leads to the following joint density of  $y_j$ ,  $t_j$ , and  $\epsilon_j$ :

$$f(y_j, t_j, \epsilon_j | \mathbf{z}_j) = f(y_j | \mathbf{z}_j, t_j, \epsilon_j) P(t_j | \mathbf{z}_j, \epsilon_j) f(\epsilon_j)$$

The density of  $y_j$  and  $t_j$ , conditioned on  $\mathbf{z}_j$ , is obtained by integrating the above with respect to  $\epsilon_j$ . Recall that  $\epsilon_j$  is normal with mean 0 and variance  $\sigma^2$ .

$$f(y_j, t_j | \mathbf{z}_j) = \int_{-\infty}^{\infty} f(y_j | \mathbf{z}_j, t_j, \epsilon_j) P(t_j | \mathbf{z}_j, \epsilon_j) \frac{1}{\sigma\sqrt{2\pi}} \exp \left\{ - \left( \frac{\epsilon_j}{\sigma\sqrt{2}} \right)^2 \right\} d\epsilon_j$$

$f(y_j, t_j | \mathbf{z}_j)$  cannot be evaluated in a closed form. We change the variable of integration from  $\epsilon_j$  to  $\eta_j = \epsilon_j / (\sigma\sqrt{2})$ , which yields

$$f(y_j, t_j | \mathbf{z}_j) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} f(y_j | \mathbf{z}_j, t_j, \sqrt{2}\sigma\eta_j) P(t_j | \mathbf{z}_j, \sqrt{2}\sigma\eta_j) \exp(-\eta_j^2) d\eta_j$$

We approximate this integral by Gauss–Hermite quadrature. Observing a sample of  $t_j$ ,  $y_j$ , and  $\mathbf{z}_j$ , we calculate the log likelihood as the following:

$$\ln L = \sum_{j=1}^n w_j \ln \{f(y_j, t_j | \mathbf{z}_j)\}$$

The  $w_j$  terms denote optional weights.

In the maximum likelihood estimation,  $\sigma$  and  $\rho$  are not directly estimated. Directly estimated are  $\ln \sigma$  and  $\operatorname{atanh} \rho$ :

$$\operatorname{atanh} \rho = \frac{1}{2} \ln \left( \frac{1 + \rho}{1 - \rho} \right)$$

Now we present formulas for the ATE and ATET. First, we nest the endogenous-treatment Poisson regression model in a potential-outcome model. A potential-outcome model specifies what each individual would obtain in each treatment level.

A potential-outcome model that nests the endogenous-treatment Poisson regression fit by **etpoisson** is

$$\begin{aligned} E(y_{0j} | \mathbf{x}_j, \epsilon_j) &= \exp(\mathbf{x}_j \boldsymbol{\beta}_0 + \epsilon_{0j}) \\ E(y_{1j} | \mathbf{x}_j, \epsilon_j) &= \exp(\mathbf{x}_j \boldsymbol{\beta}_1 + \epsilon_{1j}) \\ t_j &= \begin{cases} 1, & \text{if } \mathbf{w}_j \boldsymbol{\gamma} + u_j > 0 \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

where  $y_{0j}$  is the outcome that person  $j$  obtains if person  $j$  selects treatment 0, and  $y_{1j}$  is the outcome that person  $j$  obtains if person  $j$  selects treatment 1. This formulation allows differing coefficients for the control ( $\boldsymbol{\beta}_0$ ) and treatment ( $\boldsymbol{\beta}_1$ ) regimes. The constant intercept for the control group is  $\beta_{00}$ . The constant intercept for the treatment group is  $\beta_{11} = \beta_{00} + \delta$ , where  $\delta$  is the coefficient for treatment  $t_j$  in the outcome (1). The remaining notation was defined above.

We may allow other coefficients to differ across regimes in the outcome (1) by adding interactions between the treatment  $t_j$  and covariates  $\mathbf{x}_j$  to the model. To be concise, we use two coefficient vectors  $\boldsymbol{\beta}_0$  and  $\boldsymbol{\beta}_1$  here rather than a single coefficient vector with interactions between the treatment  $t_j$  and covariates  $\mathbf{x}_j$ . The two formulations are equivalent.

We never observe both  $y_{0j}$  and  $y_{1j}$ , only one or the other. We observe

$$y_j = t_j y_{1j} + (1 - t_j) y_{0j}$$

The vector of error terms  $(\epsilon_{0j}, \epsilon_{1j}, u_j)'$  comes from a mean zero trivariate normal distribution with covariance matrix

$$\begin{bmatrix} \sigma^2 & \theta & \sigma\rho \\ \theta & \sigma^2 & \sigma\rho \\ \sigma\rho & \sigma\rho & 1 \end{bmatrix}$$

The parameters  $\sigma$  and  $\rho$  were discussed earlier. The parameter  $\theta$  is the covariance between the two potential outcomes. We cannot estimate  $\theta$  because we have no observations in which an individual is observed in both potential outcomes. Fortunately,  $\theta$  is not required for the calculations that we present.

The ATE is the difference in means of the potential outcomes. The mean of each potential outcome accounts for each individual's contribution, regardless of whether that individual selects that treatment level.

The conditional means of the potential outcomes  $y_{tj}$ ,  $t \in (0, 1)$  for exogenous covariates  $\mathbf{z}_j$  are

$$E(y_{tj}|\mathbf{z}_j) = \exp\left(\mathbf{x}_j\boldsymbol{\beta}_t + \frac{\sigma^2}{2}\right)$$

We can see that when the coefficients are the same across the regimes, the ratio of potential-outcome means will be equal to  $\exp(\delta)$ ; this is true of the conditional and marginal potential-outcome means.

The difference in potential-outcome means or treatment effect at exogenous covariates  $\mathbf{z}_j$  is

$$E(y_{1j} - y_{0j}|\mathbf{z}_j) = \{\exp(\mathbf{x}_j\boldsymbol{\beta}_1) - \exp(\mathbf{x}_j\boldsymbol{\beta}_0)\} \exp\left(\frac{\sigma^2}{2}\right)$$

By the law of iterated expectations, the ATE is

$$\begin{aligned} E(y_{1j} - y_{0j}) &= E\{E(y_{1j} - y_{0j}|\mathbf{z}_j)\} \\ &= E\left[\{\exp(\mathbf{x}_j\boldsymbol{\beta}_1) - \exp(\mathbf{x}_j\boldsymbol{\beta}_0)\} \exp\left(\frac{\sigma^2}{2}\right)\right] \end{aligned}$$

This expectation can be estimated as a predictive margin.

Now we will derive an expression for the ATET.

The conditional means of the potential outcomes  $y_{tj}$ ,  $t \in (0, 1)$  for exogenous covariates  $\mathbf{z}_j$  and treatment  $t_j$  are

$$E(y_{tj}|\mathbf{z}_j, t_j) = \exp\left(\mathbf{x}_j\boldsymbol{\beta}_t + \frac{\sigma^2}{2}\right) \left\{\frac{\Phi(\rho\sigma + \mathbf{w}_j\boldsymbol{\gamma})}{\Phi(\mathbf{w}_j\boldsymbol{\gamma})}\right\}^{t_j} \left\{\frac{1 - \Phi(\rho\sigma + \mathbf{w}_j\boldsymbol{\gamma})}{1 - \Phi(\mathbf{w}_j\boldsymbol{\gamma})}\right\}^{1-t_j}$$

Rather than the conditional potential-outcome means, the conditional mean of the observed outcome may be of interest. The conditional mean of the observed outcome  $y_j$  for endogenous treatment indicator  $t_j$  and exogenous covariates  $\mathbf{z}_j$  is given by

$$\begin{aligned} E(y_j|\mathbf{z}_j, t_j) &= t_j \exp\left(\mathbf{x}_j\boldsymbol{\beta}_1 + \frac{\sigma^2}{2}\right) \frac{\Phi(\rho\sigma + \mathbf{w}_j\boldsymbol{\gamma})}{\Phi(\mathbf{w}_j\boldsymbol{\gamma})} \\ &\quad + (1 - t_j) \exp\left(\mathbf{x}_j\boldsymbol{\beta}_0 + \frac{\sigma^2}{2}\right) \frac{1 - \Phi(\rho\sigma + \mathbf{w}_j\boldsymbol{\gamma})}{1 - \Phi(\mathbf{w}_j\boldsymbol{\gamma})} \end{aligned}$$

The treatment effect at exogenous covariates  $\mathbf{z}_j$  and treatment  $t_j$  is

$$E(y_{1j} - y_{0j} | \mathbf{z}_j, t_j) = \{ \exp(\mathbf{x}_j \beta_1) - \exp(\mathbf{x}_j \beta_0) \} \exp\left(\frac{\sigma^2}{2}\right) \left\{ \frac{\Phi(\rho\sigma + \mathbf{w}'_j \gamma)}{\Phi(\mathbf{w}_j \gamma)} \right\}^{t_j} \left\{ \frac{1 - \Phi(\rho\sigma + \mathbf{w}_j \gamma)}{1 - \Phi(\mathbf{w}'_j \gamma)} \right\}^{1-t_j}$$

By the law of iterated expectations, the ATET is

$$E(y_{1j} - y_{0j} | t_j = 1) = E\{E(y_{1j} - y_{0j} | \mathbf{z}_j, t_j = 1) | t_j = 1\} \\ = E\left[ \{ \exp(\mathbf{x}_j \beta_1) - \exp(\mathbf{x}_j \beta_0) \} \exp\left(\frac{\sigma^2}{2}\right) \frac{\Phi(\rho\sigma + \mathbf{w}_j \gamma)}{\Phi(\mathbf{w}_j \gamma)} \Big| t_j = 1 \right]$$

This can be estimated as a predictive margin on the treated subpopulation.

We note that when  $\rho = 0$ , the correction factor involving  $\Phi$  will disappear from the ATET. Then the ATE and ATET will be equivalent if the distribution of  $\mathbf{x}_j$  under the treated population is identical to the distribution over the entire population.

The probability of  $y_j$  conditional on  $t_j$  and  $\mathbf{z}_j$  is

$$\Pr(y_j = n | \mathbf{z}_j, t_j) = \frac{f(y_j = n, t_j | \mathbf{z}_j)}{\Phi(\mathbf{w}_j \gamma)^{t_j} \Phi(-\mathbf{w}_j \gamma)^{1-t_j}}$$

As discussed earlier, we approximate  $f(y_j, t_j | \mathbf{z}_j)$  using Gauss–Hermite quadrature.

## References

- Cerulli, G. 2015. *Econometric Evaluation of Socio-Economic Programs: Theory and Applications*. Berlin: Springer.
- Hasebe, T. 2020. Endogenous switching regression model and treatment effects of count-data outcome. *Stata Journal* 20: 627–646.
- Terza, J. V. 1998. Estimating count data models with endogenous switching: Sample selection and endogenous treatment effects. *Journal of Econometrics* 84: 129–154. [https://doi.org/10.1016/S0304-4076\(97\)00082-1](https://doi.org/10.1016/S0304-4076(97)00082-1).

## Also see

- [TE] **etpoisson postestimation** — Postestimation tools for etpoisson
- [TE] **etregress** — Linear regression with endogenous treatment effects
- [R] **heckpoisson** — Poisson regression with sample selection
- [R] **ivpoisson** — Poisson model with continuous endogenous covariates
- [R] **ivprobit** — Probit model with continuous endogenous covariates
- [R] **ivregress** — Single-equation instrumental-variables regression
- [R] **ivtobit** — Tobit model with continuous endogenous covariates
- [R] **poisson** — Poisson regression
- [SVY] **svy estimation** — Estimation commands for survey data
- [U] **20 Estimation and postestimation commands**

Postestimation commands  
Remarks and examples

predict  
Methods and formulas

margins  
Also see

## Postestimation commands

The following standard postestimation commands are available after `etpoisson`:

Command	Description
<code>contrast</code>	contrasts and ANOVA-style joint tests of estimates
<code>estat ic</code>	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estat (svy)</code>	postestimation statistics for survey data
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
* <code>hausman</code>	Hausman's specification test
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of coefficients
* <code>lrtest</code>	likelihood-ratio test
<code>margins</code>	marginal means, predictive margins, marginal effects, and average marginal effects
<code>marginsplot</code>	graph the results from margins (profile plots, interaction plots, etc.)
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
<code>predict</code>	potential-outcome means, observed-outcome means, conditional treatment effects, etc.
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>pwcompare</code>	pairwise comparisons of estimates
<code>suest</code>	seemingly unrelated estimation
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

\* `hausman` and `lrtest` are not appropriate with `svy` estimation results.

# predict

## Description for predict

`predict` creates a new variable containing predictions such as counts, conditional treatment effects, probabilities, and linear predictions.

## Menu for predict

Statistics > Postestimation

## Syntax for predict

```
predict [type] newvar [if] [in] [, statistic nooffset]
```

```
predict [type] stub* [if] [in], scores
```

<i>statistic</i>	Description
------------------	-------------

Main

<code>pomean</code>	potential-outcome mean (the predicted count); the default
<code>omean</code>	observed-outcome mean (the predicted count)
<code>cte</code>	conditional treatment effect at treatment level
<code>pr(<i>n</i>)</code>	probability $\Pr(y_j = n)$
<code>pr(<i>a</i>,<i>b</i>)</code>	probability $\Pr(a \leq y_j \leq b)$
<code>xb</code>	linear prediction
<code>xbtreat</code>	linear prediction for treatment equation

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

## Options for predict

Main

`pomean`, the default, calculates the potential-outcome mean.

`omean` calculates the observed-outcome mean.

`cte` calculates the treatment effect, the difference of potential-outcome means, conditioned on treatment level.

`pr(n)` calculates the probability  $\Pr(y_j = n)$ , where  $n$  is a nonnegative integer that may be specified as a number or a variable.

`pr(a,b)` calculates the probability  $\Pr(a \leq y_j \leq b)$ , where  $a$  and  $b$  are nonnegative integers that may be specified as numbers or variables;

$b$  missing ( $b \geq .$ ) means  $+\infty$ ;

`pr(20,.)` calculates  $\Pr(y_j \geq 20)$ ;

`pr(20,b)` calculates  $\Pr(y_j \geq 20)$  in observations for which  $b \geq .$  and calculates  $\Pr(20 \leq y_j \leq b)$  elsewhere.

`pr(.,b)` produces a syntax error. A missing value in an observation of the variable  $a$  causes a missing value in that observation for `pr(a,b)`.

`xb` calculates the linear prediction for the dependent count variable, which is  $\mathbf{x}_j\beta$  if neither `offset()` nor `exposure()` was specified;  $\mathbf{x}_j\beta + \text{offset}_j^\beta$  if `offset()` was specified; or  $\mathbf{x}_j\beta + \ln(\text{exposure}_j)$  if `exposure()` was specified.

`xbtreat` calculates the linear prediction for the endogenous treatment equation, which is  $\mathbf{w}_j\gamma$  if `offset()` was not specified in `treat()` and  $\mathbf{w}_j\gamma + \text{offset}_j^\alpha$  if `offset()` was specified in `treat()`.

`nooffset` is relevant only if you specified `offset()` or `exposure()` when you fit the model. It modifies the calculations made by `predict` so that they ignore the offset or exposure variable. `nooffset` removes the offset from calculations involving both the `treat()` equation and the dependent count variable.

`scores` calculates equation-level score variables.

The first new variable will contain  $\partial \ln L / \partial (\mathbf{x}_j\beta)$ .

The second new variable will contain  $\partial \ln L / \partial (\mathbf{w}_j\gamma)$ .

The third new variable will contain  $\partial \ln L / \partial \text{atanh } \rho$ .

The fourth new variable will contain  $\partial \ln L / \partial \ln \sigma$ .

## margins

### Description for margins

`margins` estimates margins of response for counts, conditional treatment effects, probabilities, and linear predictions.

### Menu for margins

Statistics > Postestimation

### Syntax for margins

```
margins [marginlist] [, options]
```

```
margins [marginlist] , predict(statistic ...) [predict(statistic ...) ...] [options]
```

<i>statistic</i>	Description
------------------	-------------

---

<code>pomean</code>	potential-outcome mean (the predicted count); the default
---------------------	---

<code>omean</code>	observed-outcome mean (the predicted count)
--------------------	---

<code>cte</code>	conditional treatment effect at treatment level
------------------	---

<code>pr(<i>n</i>)</code>	probability $\Pr(y_j = n)$
---------------------------	----------------------------

<code>pr(<i>a,b</i>)</code>	probability $\Pr(a \leq y_j \leq b)$
-----------------------------	--------------------------------------

<code>xb</code>	linear prediction
-----------------	-------------------

<code>xbtreat</code>	linear prediction for treatment equation
----------------------	--

---

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [\[R\] margins](#).

### Remarks and examples

The average treatment effect (ATE) and the average treatment effect on the treated (ATET) are the parameters most frequently estimated by postestimation techniques after `etpoisson`.

You can use the `margins` command (see [\[R\] margins](#)) after `etpoisson` to estimate the ATE or ATET. See [example 2](#) of [\[TE\] etpoisson](#) for an example of ATE estimation. See [example 3](#) of [\[TE\] etpoisson](#) for an example of ATET estimation.

See [example 1](#) of [\[TE\] etpoisson](#) for an example using `lincom` after `etpoisson`.

### Methods and formulas

See [Methods and formulas](#) of [\[TE\] etpoisson](#) for details.

### Also see

[\[TE\] etpoisson](#) — Poisson regression with endogenous treatment effects

[\[U\] 20 Estimation and postestimation commands](#)



# Title

**etregress** — Linear regression with endogenous treatment effects

[Description](#)

[Menu](#)

[Options for maximum likelihood estimates](#)

[Options for control-function estimates](#)

[Stored results](#)

[References](#)

[Quick start](#)

[Syntax](#)

[Options for two-step consistent estimates](#)

[Remarks and examples](#)

[Methods and formulas](#)

[Also see](#)

## Description

`etregress` estimates an average treatment effect (ATE) and the other parameters of a linear regression model augmented with an endogenous binary-treatment variable. Estimation is by full maximum likelihood, a two-step consistent estimator, or a control-function estimator.

In addition to the ATE, `etregress` can be used to estimate the average treatment effect on the treated (ATET) when the outcome may not be conditionally independent of the treatment.

## Quick start

ATE and ATET from a linear regression model of  $y$  on  $x$  and endogenous binary treatment `treat` modeled by  $x$  and  $w$

```
etregress y x, treat(treat = x w)
```

As above, but use a control-function estimator

```
etregress y x, treat(treat = x w) cfunction
```

With robust standard errors

```
etregress y x, treat(treat = x w) vce(robust)
```

Add the interaction between `treat` and continuous covariate  $x$  using [factor variables](#)

```
etregress y x i.treat#c.x, treat(treat = x w) vce(robust)
```

ATE after `etregress` with the required `vce(robust)` option and endogenous treatment interaction terms

```
margins r.treat, vce(unconditional)
```

As above, but calculate ATET

```
margins, vce(unconditional) predict(cte) subpop(if treat==1)
```

## Menu

Statistics > Treatment effects > Endogenous treatment > Maximum likelihood estimator > Continuous outcomes

## Syntax

*Basic syntax*

```
etregress depvar [indepvars], treat(depvart = indepvarst) [twostep|cfunction]
```

*Full syntax for maximum likelihood estimates only*

```
etregress depvar [indepvars] [if] [in] [weight],  
treat(depvart = indepvarst [, noconstant]) [etregress_ml_options]
```

*Full syntax for two-step consistent estimates only*

```
etregress depvar [indepvars] [if] [in],  
treat(depvart = indepvarst [, noconstant]) twostep [etregress_ts_options]
```

*Full syntax for control-function estimates only*

```
etregress depvar [indepvars] [if] [in],  
treat(depvart = indepvarst [, noconstant]) cfunction [etregress_cf_options]
```

<i>etregress_ml_options</i>	Description
Model	
* <u>treat</u> ()	equation for treatment effects
<u>noconstant</u>	suppress constant term
<u>putcomes</u>	use potential-outcome model with separate treatment and control group variance and correlation parameters
<u>constraints</u> ( <i>constraints</i> )	apply specified linear constraints
SE/Robust	
<u>vce</u> ( <i>vcetype</i> )	<i>vcetype</i> may be <u>oim</u> , <u>robust</u> , <u>cluster</u> <i>clustvar</i> , <u>opg</u> , <u>bootstrap</u> , or <u>jackknife</u>
Reporting	
<u>level</u> (#)	set confidence level; default is <u>level</u> (95)
<u>first</u>	report first-step probit estimates
<u>hazard</u> ( <i>newvar</i> )	create <i>newvar</i> containing hazard from treatment equation
<u>lrmmodel</u>	perform the likelihood-ratio model test instead of the default Wald test
<u>nocnsreport</u>	do not display constraints
<u>display_options</u>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<u>maximize_options</u>	control the maximization process; seldom used
<u>collinear</u>	keep collinear variables
<u>coeflegend</u>	display legend instead of statistics

\*treat(*depar*<sub>*t*</sub> = *indepvars*<sub>*t*</sub> [ , noconstant ]) is required.

<i>etregress_ts_options</i>	Description
Model	
* <u>treat</u> ()	equation for treatment effects
* <u>twostep</u>	produce two-step consistent estimate
<u>noconstant</u>	suppress constant term
SE	
<u>vce</u> ( <i>vcetype</i> )	<i>vcetype</i> may be <u>conventional</u> , <u>bootstrap</u> , or <u>jackknife</u>
Reporting	
<u>level</u> (#)	set confidence level; default is <u>level</u> (95)
<u>first</u>	report first-step probit estimates
<u>hazard</u> ( <i>newvar</i> )	create <i>newvar</i> containing hazard from treatment equation
<u>display_options</u>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
<u>coeflegend</u>	display legend instead of statistics

\*treat(*depar*<sub>*t*</sub> = *indepvars*<sub>*t*</sub> [ , noconstant ]) and twostep are required.

<i>etregress_cf_options</i>	Description
Model	
* <b>treat()</b>	equation for treatment effects
* <b>cfunction</b>	produce control-function estimate
<b>noconstant</b>	suppress constant term
<b>putcomes</b>	use potential-outcome model with separate treatment and control group variance and correlation parameters
SE	
<b>vce(vctype)</b>	<i>vctype</i> may be <b>robust</b> , <b>bootstrap</b> , or <b>jackknife</b>
Reporting	
<b>level(#)</b>	set confidence level; default is <b>level(95)</b>
<b>first</b>	report first-step probit estimates
<b>hazard(newvar)</b>	create <i>newvar</i> containing hazard from treatment equation
<b>display_options</b>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<b>maximize_options</b>	control the maximization process; seldom used
<b>coeflegend</b>	display legend instead of statistics

\***treat**(*depar*<sub>*t*</sub> = *indepvars*<sub>*t*</sub> [ , **noconstant** ]) and **cfunction** are required.

*indepvars* and *indepvars*<sub>*t*</sub> may contain factor variables; see [U] 11.4.3 **Factor variables**.

*depar*, *indepvars*, *depar*<sub>*t*</sub>, and *indepvars*<sub>*t*</sub> may contain time-series operators; see [U] 11.4.4 **Time-series varlists**.

**bootstrap**, **by**, **collect**, **fp**, **jackknife**, **rolling**, **statsby**, and **svy** are allowed; see [U] 11.1.10 **Prefix commands**.

Weights are not allowed with the **bootstrap** prefix; see [R] **bootstrap**.

**aweight**s are not allowed with the **jackknife** prefix; see [R] **jackknife**.

**twostep**, **cfunction**, **vce()**, **first**, **hazard()**, **lrmmodel**, and **weights** are not allowed with the **svy** prefix; see [SVY] **svy**.

**pweight**s, **aweight**s, **fweight**s, and **iweight**s are allowed with both maximum likelihood and control-function estimation; see [U] 11.1.6 **weight**. No weights are allowed if **twostep** is specified.

**collinear** and **coeflegend** do not appear in the dialog box.

See [U] 20 **Estimation and postestimation commands** for more capabilities of estimation commands.

## Options for maximum likelihood estimates

### Model

`treat(depcvart = indepvarst [ , noconstant ])` specifies the variables and options for the treatment equation. It is an integral part of specifying a treatment-effects model and is required.

`noconstant`; see [R] [Estimation options](#).

`poutcomes` specifies that a potential-outcome model with separate variance and correlation parameters for each of the treatment and control groups be used.

`constraints(constraints)`; see [R] [Estimation options](#).

### SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`oim`, `opg`), that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster clustvar`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce\\_option](#).

### Reporting

`level(#)`; see [R] [Estimation options](#).

`first` specifies that the first-step probit estimates of the treatment equation be displayed before estimation.

`hazard(newvar)` will create a new variable containing the hazard from the treatment equation. The hazard is computed from the estimated parameters of the treatment equation.

`lrmodel`, `nocnsreport`; see [R] [Estimation options](#).

`display_options`: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

### Maximization

`maximize_options`: `difficult`, `technique(algorithm_spec)`, `iterate(#)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, `nonrtolerance`, and `from(init_specs)`; see [R] [Maximize](#). These options are seldom used.

Setting the optimization type to `technique(bhhh)` resets the default `vcetype` to `vce(opg)`.

The following options are available with `etregress` but are not shown in the dialog box:

`collinear`, `coeflegend`; see [R] [Estimation options](#).

## Options for two-step consistent estimates

### Model

`treat(depcvart = indepvarst [ , noconstant ])` specifies the variables and options for the treatment equation. It is an integral part of specifying a treatment-effects model and is required.

`twostep` specifies that two-step consistent estimates of the parameters, standard errors, and covariance matrix be produced, instead of the default maximum likelihood estimates.

`noconstant`; see [R] [Estimation options](#).

### SE

`vce(vcetype)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`conventional`) and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce\\_option](#).

`vce(conventional)`, the default, uses the conventionally derived variance estimator for the two-step estimator of the treatment-effects model.

### Reporting

`level(#)`; see [R] [Estimation options](#).

`first` specifies that the first-step probit estimates of the treatment equation be displayed before estimation.

`hazard(newvar)` will create a new variable containing the hazard from the treatment equation. The hazard is computed from the estimated parameters of the treatment equation.

*display\_options*: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

The following option is available with `etregress` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

## Options for control-function estimates

### Model

`treat(depcvart = indepvarst [ , noconstant ])` specifies the variables and options for the treatment equation. It is an integral part of specifying a treatment-effects model and is required.

`cfunction` specifies that control-function estimates of the parameters, standard errors, and covariance matrix be produced instead of the default maximum likelihood estimates. `cfunction` is required.

`noconstant`; see [R] [Estimation options](#).

`poutcomes` specifies that a potential-outcome model with separate variance and correlation parameters for each of the treatment and control groups be used.

### SE

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`) and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce\\_option](#).

Reporting

`level(#)`; see [R] [Estimation options](#).

`first` specifies that the first-step probit estimates of the treatment equation be displayed before estimation.

`hazard(newvar)` will create a new variable containing the hazard from the treatment equation. The hazard is computed from the estimated parameters of the treatment equation.

*display\_options*: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

Maximization

*maximize\_options*: `iterate(#)`, `[no]log`, and `from(init_specs)`; see [R] [Maximize](#). These options are seldom used.

*init\_specs* is one of

`matname [ , skip copy ]`

`# [ # ... ] copy`

The following option is available with `etregress` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

## Remarks and examples

Remarks are presented under the following headings:

[Overview](#)

[Basic examples](#)

[Average treatment effect \(ATE\)](#)

[Average treatment effect on the treated \(ATET\)](#)

## Overview

`etregress` estimates an ATE and the other parameters of a linear regression model that also includes an endogenous binary-treatment variable. In addition to the ATE, the parameters estimated by `etregress` can be used to estimate the ATET when the outcome is not conditionally independent of the treatment.

We call the model fit by `etregress` an endogenous treatment-regression model, although it is also known as an endogenous binary-variable model or as an endogenous dummy-variable model. The endogenous treatment-regression model is a specific endogenous treatment-effects model; it uses a linear model for the outcome and a normal distribution to model the deviation from the conditional independence assumption imposed by the estimators implemented in `teffects`; see [TE] [teffects intro](#). In treatment-effects jargon, the endogenous binary-variable model is a linear potential-outcome model that allows for a specific correlation structure between the unobservables that affect the treatment and the unobservables that affect the potential outcomes. See [TE] [etpoisson](#) for an estimator that allows for a nonlinear outcome model and a similar model for the endogeneity of the treatment.

Heckman (1976, 1978) brought this model into the modern literature. Maddala (1983) derives the maximum likelihood and the control-function (CF) estimators of the model. Maddala (1983) also reviews some empirical applications and describes it as an endogenous-switching model. Barnow, Cain, and Goldberger (1981) provide another useful derivation of this model. They concentrate on deriving the conditions for which the self-selection bias of the simple OLS estimator of the treatment effect,  $\delta$ , is nonzero and of a specific sign. Cameron and Trivedi (2005, sec. 16.7 and 25.3.4) and Wooldridge (2010, sec. 21.4.1) discuss the endogenous binary-variable model as an endogenous treatment-effects model and link it to recent work.

`etregress` performs CF estimation in one step by using the generalized method of moments (GMM) with stacked moments. See Newey (1984) and Wooldridge (2010, sec. 14.2) for a description of this technique. Many econometric and statistical models can be expressed as conditions on the population moments. The parameter estimates produced by GMM estimators make the sample-moment conditions as true as possible given the data. See [R] `gmm` for further information on GMM estimation and how Stata performs it. Two-step CF estimation is also supported by `etregress`.

Formally, the endogenous treatment-regression model is composed of an equation for the outcome  $y_j$  and an equation for the endogenous treatment  $t_j$ . The variables  $\mathbf{x}_j$  are used to model the outcome. When there are no interactions between  $t_j$  and  $\mathbf{x}_j$ , we have

$$y_j = \mathbf{x}_j\boldsymbol{\beta} + \delta t_j + \epsilon_j$$

$$t_j = \begin{cases} 1, & \text{if } \mathbf{w}_j\boldsymbol{\gamma} + u_j > 0 \\ 0, & \text{otherwise} \end{cases}$$

where  $\mathbf{w}_j$  are the covariates used to model treatment assignment, and the error terms  $\epsilon_j$  and  $u_j$  are bivariate normal with mean zero and covariance matrix

$$\begin{bmatrix} \sigma^2 & \rho\sigma \\ \rho\sigma & 1 \end{bmatrix}$$

The covariates  $\mathbf{x}_j$  and  $\mathbf{w}_j$  are unrelated to the error terms; in other words, they are exogenous. We call this the constrained model because the variance and correlation parameters are identical across the treatment and control groups.

This model can be generalized to a potential-outcome model with separate variance and correlation parameters for the treatment and control groups. The generalized model is

$$y_{0j} = \mathbf{x}_j\boldsymbol{\beta}_0 + \epsilon_{0j}$$

$$y_{1j} = \mathbf{x}_j\boldsymbol{\beta}_1 + \epsilon_{1j}$$

$$t_j = \begin{cases} 1, & \text{if } \mathbf{w}_j\boldsymbol{\gamma} + u_j > 0 \\ 0, & \text{otherwise} \end{cases}$$

where  $y_{0j}$  is the outcome that person  $j$  obtains if person  $j$  selects treatment 0, and  $y_{1j}$  is the outcome that person  $j$  obtains if person  $j$  selects treatment 1. We never observe both  $y_{0j}$  and  $y_{1j}$ , only one or the other. We observe

$$y_j = t_j y_{1j} + (1 - t_j) y_{0j}$$



In this unconstrained model, the vector of error terms  $(\epsilon_{0j}, \epsilon_{1j}, u_j)'$  comes from a mean zero trivariate normal distribution with covariance matrix

$$\begin{bmatrix} \sigma_0^2 & \sigma_{01} & \sigma_0\rho_0 \\ \sigma_{01} & \sigma_1^2 & \sigma_1\rho_1 \\ \sigma_0\rho_0 & \sigma_1\rho_1 & 1 \end{bmatrix}$$

The covariance  $\sigma_{01}$  cannot be identified because we never observe both  $y_{1j}$  and  $y_{0j}$ . However, identification of  $\sigma_{01}$  is not necessary to estimate the other parameters because all covariates and the outcome are observed in observations from each group. We normalize the treatment error variance to be 1 because we observe only whether an outcome occurs under treatment. More details are found in *Methods and formulas*.

Rather than showing two separate regression equations, **etregress** reports one outcome equation with interaction terms between the treatment and outcome covariates. **etregress** can fit the constrained and generalized potential-outcome models using either the default maximum likelihood estimator or the one-step CF estimator obtained with option **cf**. The two-step CF estimator provides consistent estimates for the constrained model.

## Basic examples

When there are no interactions between the treatment variable and the outcome covariates in the constrained model, **etregress** directly estimates the ATE and the ATET.

### ► Example 1: Basic example

We estimate the ATE of being a union member on wages of women in 1972 from a nonrepresentative extract of the National Longitudinal Survey on young women who were ages 14–26 in 1968. We will use the variables **wage** (wage), **grade** (years of schooling completed), **smsa** (an indicator for living in an SMSA—standard metropolitan statistical area), **black** (an indicator for being African-American), **tenure** (tenure at current job), and **south** (an indicator for living in the South).

We use `etregress` to estimate the parameters of the endogenous treatment-regression model.

```
. use https://www.stata-press.com/data/r17/union3
(NLS Women 14-24 in 1968)
. etregress wage age grade smsa black tenure, treat(union = south black tenure)
Iteration 0:  log likelihood = -3140.811
Iteration 1:  log likelihood = -3053.6629
Iteration 2:  log likelihood = -3051.5847
Iteration 3:  log likelihood = -3051.575
Iteration 4:  log likelihood = -3051.575
Linear regression with endogenous treatment      Number of obs = 1,210
Estimator: Maximum likelihood                  Wald chi2(6) = 681.89
Log likelihood = -3051.575                      Prob > chi2 = 0.0000
```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
<b>wage</b>						
age	.1487409	.0193291	7.70	0.000	.1108566	.1866252
grade	.4205658	.0293577	14.33	0.000	.3630258	.4781058
smsa	.9117044	.1249041	7.30	0.000	.6668969	1.156512
black	-.7882471	.1367078	-5.77	0.000	-1.05619	-.5203048
tenure	.1524015	.0369596	4.12	0.000	.0799621	.2248409
1.union	2.945815	.2749621	10.71	0.000	2.4069	3.484731
_cons	-4.351572	.5283952	-8.24	0.000	-5.387208	-3.315936
<b>union</b>						
south	-.5807419	.0851111	-6.82	0.000	-.7475566	-.4139271
black	.4557499	.0958042	4.76	0.000	.2679771	.6435226
tenure	.0871536	.0232483	3.75	0.000	.0415878	.1327195
_cons	-.8855758	.0724506	-12.22	0.000	-1.027576	-.7435753
/athrho	-.6544347	.0910314	-7.19	0.000	-.832853	-.4760164
/lnsigma	.7026769	.0293372	23.95	0.000	.645177	.7601767
rho	-.5746478	.060971			-.682005	-.4430476
sigma	2.019151	.0592362			1.906325	2.138654
lambda	-1.1603	.1495097			-1.453334	-.8672668

```
LR test of indep. eqns. (rho = 0):  chi2(1) = 19.84  Prob > chi2 = 0.0000
```

The likelihood-ratio test in the footer indicates that we can reject the null hypothesis of no correlation between the treatment-assignment errors and the outcome errors. The estimated correlation between the treatment-assignment errors and the outcome errors,  $\rho$ , is  $-0.575$ . The negative relationship indicates that unobservables that raise observed wages tend to occur with unobservables that lower union membership. We discuss some details about this parameter in the technical note [below](#).

The estimated ATE of being a union member is 2.95. The ATET is the same as the ATE in this case because the treatment indicator variable has not been interacted with any of the outcome covariates, and the correlation and variance parameters are identical across the control and treatment groups.

## □ Technical note

The results for the ancillary parameters  $\rho$  and  $\sigma$  require explanation. For numerical stability during optimization, `etregress` does not directly estimate  $\rho$  or  $\sigma$ . Instead, `etregress` estimates the inverse hyperbolic tangent of  $\rho$ ,

$$\operatorname{atanh} \rho = \frac{1}{2} \ln \left( \frac{1 + \rho}{1 - \rho} \right)$$

and  $\ln \sigma$ . Also `etregress` reports  $\lambda = \rho \sigma$ , along with an estimate of the standard error of the estimate and the confidence interval. □

In contrast to the constrained model, `etregress` directly estimates the ATE only when there are no interactions between the treatment variable and the outcome covariates in the unconstrained model.

## ▷ Example 2: Allowing group-specific variance and correlation

We estimate the ATE of having health insurance on the natural logarithm of total out-of-pocket prescription drug expenditures from a simulated random sample of individuals between the ages of 25 and 64. We will use the variables `lndrug` (natural logarithm of spending on prescription drugs), `age` (age of the individual), `chron` (whether the individual has a chronic condition), `lninc` (natural logarithm of income), `married` (marriage status), and `work` (employment status). Our treatment is whether the person has health insurance, `ins`. We allow the outcome error variance and correlation parameters to vary between the treated (insured) and control (uninsured) groups in this example, rather than constraining them to be equal as in [example 1](#).

We use `etregress` to estimate the parameters of the endogenous treatment-effects model. To estimate separate variance and correlation parameters for each of the control and treatment groups, we specify the `poutcomes` option. We specify the `cfunction` option to use the CF estimator.

```
. use https://www.stata-press.com/data/r17/drugexp
(Prescription drug expenditures)
. etregress lndrug chron age lninc, treat(ins=age married lninc work) poutcomes
> cfunction
```

```
Iteration 0: GMM criterion Q(b) = 2.279e-15
```

```
Iteration 1: GMM criterion Q(b) = 6.358e-30
```

```
Linear regression with endogenous treatment
```

```
Number of obs = 6,000
```

```
Estimator: Control function
```

	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
<b>lndrug</b>						
chron	.4671725	.0319731	14.61	0.000	.4045064	.5298387
age	.1021359	.00292	34.98	0.000	.0964128	.1078589
lninc	.0550672	.0225036	2.45	0.014	.0109609	.0991735
1.ins	-.8598836	.3483648	-2.47	0.014	-1.542666	-.1771011
_cons	1.665539	.2527527	6.59	0.000	1.170153	2.160925
<b>ins</b>						
age	.021142	.0022961	9.21	0.000	.0166416	.0256424
married	.084631	.0359713	2.35	0.019	.0141286	.1551334
lninc	.1023032	.0225009	4.55	0.000	.0582022	.1464041
work	.288418	.0372281	7.75	0.000	.2154522	.3613837
_cons	-.622993	.108795	-5.73	0.000	-.8362273	-.4097587
<b>/athrho0</b>						
/athrho0	.4035094	.1724539	2.34	0.019	.0655059	.7415129
<b>/lnsigma0</b>						
/lnsigma0	.3159269	.0500476	6.31	0.000	.2178353	.4140184
<b>/athrho1</b>						
/athrho1	.7929459	.2986601	2.66	0.008	.2075829	1.378309
<b>/lnsigma1</b>						
/lnsigma1	.1865347	.0613124	3.04	0.002	.0663646	.3067048
<b>rho0</b>						
rho0	.3829477	.1471637			.0654124	.6300583
<b>sigma0</b>						
sigma0	1.37153	.0686418			1.243382	1.512885
<b>lambda0</b>						
lambda0	.5252243	.226367			.0815532	.9688954
<b>rho1</b>						
rho1	.6600746	.1685343			.2046518	.880572
<b>sigma1</b>						
sigma1	1.205066	.0738855			1.068616	1.35894
<b>lambda1</b>						
lambda1	.7954338	.2513036			.3028878	1.28798

```
Wald test of indep. (rho0 = rho1 = 0): chi2(2) = 8.88 Prob > chi2 = 0.0118
```

The Wald test reported in the footer indicates that we can reject the null hypothesis of no correlation between the treatment-assignment errors and the outcome errors for the control and treatment groups. The estimate of the correlation of the treatment-assignment errors for the control group ( $\rho_0$ ) is positive, indicating that unobservables that increase spending on prescription drugs tend to occur with unobservables that increase health insurance coverage. Because  $\rho_1$  is also positive, we make the same interpretation for individuals with insurance. The estimate  $\rho_1$  is larger than the estimate  $\rho_0$ , indicating a stronger relationship between the unobservables and treatment outcomes in the treated group.

The estimated ATE of having health insurance is  $-0.86$ . Note that while the ATE and ATET were the same in [example 1](#), that is not the case here. We show how to calculate the ATET for a potential-outcome model in [example 6](#).

The estimate of the outcome error standard-deviation parameter for the control group ( $\sigma_0$ ) is slightly larger than that of the treatment-group parameter ( $\sigma_1$ ), indicating a greater variability in the unobservables among the untreated group.

## Average treatment effect (ATE)

When there is a treatment variable and outcome covariate interaction, the parameter estimates from `etregress` can be used by `margins` to estimate the ATE, the average difference of the treatment potential outcomes and the control potential outcomes.

### ▷ Example 3: Allowing interactions between treatment and outcome covariates, ATE

In [example 1](#), the coefficients on the outcome covariates do not vary by treatment level. The differences in wages between union members and nonmembers are modeled as a level shift captured by the coefficient on the indicator for union membership. In this example, we use factor-variable notation to allow some of the coefficients to vary over treatment level and then use `margins` (see [\[R\] margins](#)) to estimate the ATE. (See [\[U\] 11.4.3 Factor variables](#) for an introduction to factor-variable notation.)

We begin by estimating the parameters of the model in which the coefficients on `black` and `tenure` differ for union members and nonmembers. We specify the `vce(robust)` option because we need to specify `vce(unconditional)` when we use `margins` below.

```

. use https://www.stata-press.com/data/r17/union3
(NLS Women 14-24 in 1968)
. etregress wage age grade smsa i.union#c.(black tenure),
> treat(union = south black tenure) vce(robust)

Iteration 0:  log pseudolikelihood = -3614.6714
Iteration 1:  log pseudolikelihood = -3218.8152
Iteration 2:  log pseudolikelihood = -3057.0115
Iteration 3:  log pseudolikelihood = -3049.3081
Iteration 4:  log pseudolikelihood = -3049.2838
Iteration 5:  log pseudolikelihood = -3049.2838

Linear regression with endogenous treatment          Number of obs = 1,210
Estimator: Maximum likelihood                      Wald chi2(8) = 493.40
Log pseudolikelihood = -3049.2838                 Prob > chi2 = 0.0000

```

	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
<b>wage</b>						
age	.1489075	.0207283	7.18	0.000	.1082809	.1895342
grade	.4200493	.0377621	11.12	0.000	.3460371	.4940616
smsa	.9232615	.1201486	7.68	0.000	.6877746	1.158748
union#						
c.black						
0	-.6685582	.1444213	-4.63	0.000	-.9516187	-.3854977
1	-1.1831	.2574817	-4.59	0.000	-1.687755	-.6784455
union#						
c.tenure						
0	.168746	.0503107	3.35	0.001	.0701388	.2673532
1	.0836367	.0903669	0.93	0.355	-.0934792	.2607526
1.union	3.342859	.5586863	5.98	0.000	2.247854	4.437864
_cons	-4.42566	.6493003	-6.82	0.000	-5.698265	-3.153055
<b>union</b>						
south	-.5844678	.0833069	-7.02	0.000	-.7477464	-.4211893
black	.4740688	.093241	5.08	0.000	.2913197	.6568178
tenure	.0874297	.0253892	3.44	0.001	.0376678	.1371916
_cons	-.8910484	.0746329	-11.94	0.000	-1.037326	-.7447706
/athrho	-.6733149	.2215328	-3.04	0.002	-1.107511	-.2391185
/lnsigma	.7055907	.0749711	9.41	0.000	.55865	.8525313
rho	-.5871562	.1451589			-.8031809	-.234663
sigma	2.025042	.1518197			1.748311	2.345577
lambda	-1.189016	.3631079			-1.900695	-.4773378

Wald test of indep. eqns. (rho = 0): chi2(1) = 9.24 Prob > chi2 = 0.0024

The results indicate that the coefficients on **black** differ by union membership and that the coefficient on **tenure** for nonmembers is positive, while the coefficient on **tenure** for members is 0. The model fits well overall, so we proceed with interpretation. Because we interacted the treatment variable with two of the covariates, the estimated coefficient on the treatment level is not an estimate of the ATE. Below we use **margins** to estimate the ATE from these results. We specify the **vce(unconditional)** option to obtain the standard errors for the population ATE instead of the sample ATE. We specify the **contrast(nowald)** option to suppress the Wald tests, which **margins** displays by default for contrasts.

```
. margins r.union, vce(unconditional) contrast(nowald)
Contrasts of predictive margins                                Number of obs = 1,210
Expression: Linear prediction, predict()
```

	Unconditional			
	Contrast	std. err.	[95% conf. interval]	
union (1 vs 0)	3.042688	.5305151	2.002898	4.082478

The ATE estimate is essentially the same as the one produced by the constrained model in [example 1](#). ◀

We can use the same methods above to obtain the ATE in an unconstrained model.

▷ Example 4: Treatment interactions and group-specific variance and correlation, ATE

In [example 2](#), the coefficients on the outcome covariates do not vary by treatment level. Suppose we believe that the effect of having a chronic condition on out-of-pocket spending differs between the insured and uninsured. Again, we use an interaction term. Because we are using a CF estimator, the variance–covariance of the estimator (VCE) is already robust so we do not specify `vce(robust)`.

```

. use https://www.stata-press.com/data/r17/drugexp
(Prescription drug expenditures)
. etregress lndrug i.ins#i.chron age lninc, treat(ins=age married lninc work)
> poutcomes cfunction

Iteration 0:   GMM criterion Q(b) = 2.279e-15
Iteration 1:   GMM criterion Q(b) = 1.561e-28

Linear regression with endogenous treatment          Number of obs = 6,000
Estimator: Control function

```

	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
<b>lndrug</b>						
ins#chron						
0 1	.3798705	.0720713	5.27	0.000	.2386132	.5211277
1 1	.4957773	.0352571	14.06	0.000	.4266746	.5648801
age	.1022045	.0029228	34.97	0.000	.0964758	.1079331
lninc	.0548917	.0225219	2.44	0.015	.0107497	.0990337
1.ins	-.89703	.3493058	-2.57	0.010	-1.581657	-.2124031
_cons	1.691336	.2531222	6.68	0.000	1.195225	2.187446
<b>ins</b>						
age	.021142	.0022961	9.21	0.000	.0166416	.0256424
married	.084631	.0359713	2.35	0.019	.0141286	.1551334
lninc	.1023032	.0225009	4.55	0.000	.0582022	.1464041
work	.288418	.0372281	7.75	0.000	.2154522	.3613837
_cons	-.622993	.108795	-5.73	0.000	-.8362273	-.4097587
<b>/athrho0</b>						
/athrho0	.4046007	.1725597	2.34	0.019	.0663899	.7428115
<b>/lnsigma0</b>						
/lnsigma0	.3157561	.0501956	6.29	0.000	.2173746	.4141376
<b>/athrho1</b>						
/athrho1	.7950592	.2992825	2.66	0.008	.2084763	1.381642
<b>/lnsigma1</b>						
/lnsigma1	.1868903	.0614281	3.04	0.002	.0664934	.3072871
<b>rho0</b>						
rho0	.3838786	.1471308			.0662925	.6308408
<b>sigma0</b>						
sigma0	1.371296	.0688329			1.24281	1.513065
<b>lambda0</b>						
lambda0	.5264111	.2264197			.0826366	.9701856
<b>rho1</b>						
rho1	.6612655	.1684146			.2055076	.8813184
<b>sigma1</b>						
sigma1	1.205495	.0740512			1.068754	1.359731
<b>lambda1</b>						
lambda1	.7971523	.2514293			.3043599	1.289945

Wald test of indep. (rho0 = rho1 = 0): chi2(2) = 8.90 Prob > chi2 = 0.0117

The results indicate that the coefficient on `chron` differs by whether an individual has insurance. The model fits well overall, so we proceed with interpretation.

Because we interacted the treatment variable with one of the covariates, the estimated coefficient on the treatment level is not an estimate of the ATE. Below we use `margins` to estimate the ATE from these results. We specify the `vce(unconditional)` option to obtain the standard errors for the population ATE instead of the sample ATE. We specify the `contrast(nowald)` option to suppress the Wald tests.



```
. margins r.ins, vce(unconditional) contrast(nowald)
Contrasts of predictive margins                                Number of obs = 6,000
Expression: Linear prediction, predict()
```

	Unconditional			
	Contrast	std. err.	[95% conf. interval]	
ins (1 vs 0)	-.8632045	.3484924	-1.546237	-.1801718

The ATE estimate is similar to the one produced by the constrained model in [example 2](#).



### Average treatment effect on the treated (ATET)

When there is a treatment variable and outcome covariate interaction, the parameter estimates from `etregress` can be used by `margins` to estimate the ATET, the average difference of the treatment potential outcomes and the control potential outcomes on the treated population.

#### ► Example 5: Allowing interactions between treatment and outcome covariates, ATET

The ATET may differ from the ATE in [example 3](#) because the interaction between the treatment variable and some outcome covariates makes the ATE and the ATET vary over outcome covariate values. Below we use `margins` to estimate the ATET by specifying the `subpop(union)` option, which restricts the sample used by `margins` to union members.

```
. use https://www.stata-press.com/data/r17/union3
(NLS Women 14-24 in 1968)
. etregress wage age grade smsa i.union#c.(black tenure),
> treat(union = south black tenure) vce(robust)
(output omitted)
. margins r.union, vce(unconditional) contrast(nowald) subpop(union)
Contrasts of predictive margins                                Number of obs = 1,210
Subpop. no. obs = 253
Expression: Linear prediction, predict()
```

	Unconditional			
	Contrast	std. err.	[95% conf. interval]	
union (1 vs 0)	2.968977	.5358457	1.918739	4.019215

The estimated ATET and ATE are close, indicating that the average predicted outcome for the treatment group is similar to the average predicted outcome for the whole population.



## ▷ Example 6: Treatment interactions and group-specific variance and correlation, ATET

The ATET may differ from the ATE in [example 4](#) because the interaction between the treatment variable and some outcome covariates makes the ATE and the ATET vary over values of the covariate in the outcome equation. Even if there is no interaction between treatment assignment and a covariate in the outcome equation, the estimated ATE and ATET will differ if the variances of the outcome errors and their correlations with the treatment-assignment errors differ across the control and treatment groups.

We can estimate the ATET of having health insurance by using the conditional treatment effect (conditional on exogenous covariates and treatment level) obtained using the `predict`, `cte` and the `margins` commands; see [Methods and formulas](#) below and [\[TE\] etregress postestimation](#) for more details about the use of `predict` after `etregress`.

We restrict estimation to the treated subpopulation by specifying the `subpop(ins)` option with `margins`.

```
. use https://www.stata-press.com/data/r17/drugexp
(Prescription drug expenditures)
. etregress lndrug i.ins#i.chron age lninc,
> treat(ins = age married lninc work) poutcomes cfunction
(output omitted)
. margins, predict(cte) subpop(ins) vce(unconditional)
Predictive margins                                Number of obs   = 6,000
                                                Subpop. no. obs = 4,556
Expression: Conditional treatment effect, predict(cte)
```

	Unconditional				[95% conf. interval]	
	Margin	std. err.	z	P> z		
_cons	-.7558373	.3827579	-1.97	0.048	-1.506029	-.0056457

In absolute value, the treatment effect on the treated of  $-0.76$  is smaller than the population average effect of  $-0.86$  that we found in [example 4](#).

## Stored results

etregress (maximum likelihood) stores the following in `e()`:

### Scalars

<code>e(N)</code>	number of observations
<code>e(k)</code>	number of parameters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_eq_model)</code>	number of equations in overall model test
<code>e(k_aux)</code>	number of auxiliary parameters
<code>e(k_dv)</code>	number of dependent variables
<code>e(df_m)</code>	model degrees of freedom
<code>e(ll)</code>	log likelihood
<code>e(ll_0)</code>	log likelihood, constant-only model ( <code>lrmodel</code> only)
<code>e(N_clust)</code>	number of clusters
<code>e(lambda)</code>	estimate of $\lambda$ in constrained model
<code>e(selambda)</code>	standard error of $\lambda$ in constrained model
<code>e(sigma)</code>	estimate of $\sigma$ in constrained model
<code>e(lambda0)</code>	estimate of $\lambda_0$ in potential-outcome model
<code>e(selambda0)</code>	standard error of $\lambda_0$ in potential-outcome model
<code>e(sigma0)</code>	estimate of $\sigma_0$ in potential-outcome model
<code>e(lambda1)</code>	estimate of $\lambda_1$ in potential-outcome model
<code>e(selambda1)</code>	standard error of $\lambda_1$ in potential-outcome model
<code>e(sigma1)</code>	estimate of $\sigma_1$ in potential-outcome model
<code>e(chi2)</code>	$\chi^2$
<code>e(chi2_c)</code>	$\chi^2$ for comparison test
<code>e(p)</code>	$p$ -value for model test
<code>e(p_c)</code>	$p$ -value for comparison test
<code>e(rho)</code>	estimate of $\rho$ in constrained model
<code>e(rho0)</code>	estimate of $\rho_0$ in potential-outcome model
<code>e(rho1)</code>	estimate of $\rho_1$ in potential-outcome model
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(rank0)</code>	rank of <code>e(V)</code> for constant-only model
<code>e(ic)</code>	number of iterations
<code>e(rc)</code>	return code
<code>e(converged)</code>	1 if converged, 0 otherwise

### Macros

<code>e(cmd)</code>	<code>etregress</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(hazard)</code>	variable containing hazard
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(title2)</code>	secondary title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(chi2type)</code>	Wald or LR; type of model $\chi^2$ test
<code>e(chi2_ct)</code>	Wald or LR; type of model $\chi^2$ test corresponding to <code>e(chi2_c)</code>
<code>e(vce)</code>	<i>vctype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(opt)</code>	type of optimization
<code>e(which)</code>	max or min; whether optimizer is to perform maximization or minimization
<code>e(method)</code>	m1
<code>e(ml_method)</code>	type of m1 method
<code>e(user)</code>	name of likelihood-evaluator program
<code>e(technique)</code>	maximization technique
<code>e(properties)</code>	b V
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(footnote)</code>	program used to implement the footnote display
<code>e(marginsok)</code>	predictions allowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices	
e(b)	coefficient vector
e(Cns)	constraints matrix
e(iolog)	iteration log (up to 20 iterations)
e(gradient)	gradient vector
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, $p$ -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r`-class command is run after the estimation command.

**etregress** (two-step) stores the following in `e()`:

Scalars	
e(N)	number of observations
e(df_m)	model degrees of freedom
e(lambda)	$\lambda$
e(selambda)	standard error of $\lambda$
e(sigma)	estimate of sigma
e(chi2)	$\chi^2$
e(p)	$p$ -value for model test
e(rho)	$\rho$
e(rank)	rank of <code>e(V)</code>
Macros	
e(cmd)	<b>etregress</b>
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(hazard)	variable containing hazard
e(title)	title in estimation output
e(title2)	secondary title in estimation output
e(chi2type)	Wald or LR; type of model $\chi^2$ test
e(vce)	<i>vctype</i> specified in <code>vce()</code>
e(method)	<b>twostep</b>
e(properties)	<b>b V</b>
e(predict)	program used to implement <b>predict</b>
e(footnote)	program used to implement the footnote display
e(marginsok)	predictions allowed by <b>margins</b>
e(marginsnotok)	predictions disallowed by <b>margins</b>
e(asbalanced)	factor variables <b>fvset</b> as <b>asbalanced</b>
e(asobserved)	factor variables <b>fvset</b> as <b>asobserved</b>

Matrices	
e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, $p$ -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r`-class command is run after the estimation command.

`etregress` (control-function) stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(k)</code>	number of parameters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_aux)</code>	number of auxiliary parameters
<code>e(k_dv)</code>	number of dependent variables
<code>e(lambda)</code>	estimate of $\lambda$ in constrained model
<code>e(selambda)</code>	standard error of $\lambda$ in constrained model
<code>e(sigma)</code>	estimate of $\sigma$ in constrained model
<code>e(lambda0)</code>	estimate of $\lambda_0$ in potential-outcome model
<code>e(selambda0)</code>	standard error of $\lambda_0$ in potential-outcome model
<code>e(sigma0)</code>	estimate of $\sigma_0$ in potential-outcome model
<code>e(lambda1)</code>	estimate of $\lambda_1$ in potential-outcome model
<code>e(selambda1)</code>	standard error of $\lambda_1$ in potential-outcome model
<code>e(sigma1)</code>	estimate of $\sigma_1$ in potential-outcome model
<code>e(chi2_c)</code>	$\chi^2$ for comparison test
<code>e(p_c)</code>	$p$ -value for comparison test
<code>e(rho)</code>	estimate of $\rho$ in constrained model
<code>e(rho0)</code>	estimate of $\rho_0$ in potential-outcome model
<code>e(rho1)</code>	estimate of $\rho_1$ in potential-outcome model
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	<code>etregress</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(hazard)</code>	variable containing hazard
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(title2)</code>	secondary title in estimation output
<code>e(chi2_ct)</code>	Wald; type of model $\chi^2$ test corresponding to <code>e(chi2_c)</code>
<code>e(vce)</code>	<code>vcetype</code> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(method)</code>	<code>cfunction</code>
<code>e(properties)</code>	<code>b V</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(footnote)</code>	program used to implement the footnote display
<code>e(marginsok)</code>	predictions allowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance-covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, $p$ -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r`-class command is run after the estimation command.

## Methods and formulas

Maddala (1983, 117–122 and 223–228) derives both the maximum likelihood and the CF estimators implemented here. Greene (2012, 890–894) also provides an introduction to the treatment-effects model. Cameron and Trivedi (2005, sections 16.7 and 25.3.4) and Wooldridge (2010, section 21.4.1) discuss the endogenous binary-variable model as an endogenous treatment-effects model and link it to recent work.

Methods and formulas are presented under the following headings:

*Constrained model*  
*General potential-outcome model*  
*Average treatment effect*  
*Average treatment effect on the treated*

### Constrained model

The primary regression equation of interest is

$$y_j = \mathbf{x}_j\boldsymbol{\beta} + \delta t_j + \epsilon_j \quad (1)$$

where  $t_j$  is a binary-treatment variable that is assumed to stem from an unobservable latent variable:

$$t_j^* = \mathbf{w}_j\boldsymbol{\gamma} + u_j$$

The decision to obtain the treatment is made according to the rule

$$t_j = \begin{cases} 1, & \text{if } t_j^* > 0 \\ 0, & \text{otherwise} \end{cases}$$

where  $\epsilon$  and  $u$  are bivariate normal with mean zero and covariance matrix

$$\begin{bmatrix} \sigma^2 & \rho\sigma \\ \rho\sigma & 1 \end{bmatrix}$$

Interactions between  $\mathbf{x}_j$  and the treatment  $t_j$  are also allowed in (1). The likelihood function for this model is given in Maddala (1983, 122). Greene (2000, 180) discusses the standard method of reducing a bivariate normal to a function of a univariate normal and the correlation  $\rho$ . The following is the log likelihood for observation  $j$ ,

$$\ln L_j = \begin{cases} \ln\Phi\left\{\frac{\mathbf{w}_j\boldsymbol{\gamma} + (y_j - \mathbf{x}_j\boldsymbol{\beta} - \delta)\rho/\sigma}{\sqrt{1 - \rho^2}}\right\} - \frac{1}{2}\left(\frac{y_j - \mathbf{x}_j\boldsymbol{\beta} - \delta}{\sigma}\right)^2 - \ln(\sqrt{2\pi}\sigma) & t_j = 1 \\ \ln\Phi\left\{\frac{-\mathbf{w}_j\boldsymbol{\gamma} - (y_j - \mathbf{x}_j\boldsymbol{\beta})\rho/\sigma}{\sqrt{1 - \rho^2}}\right\} - \frac{1}{2}\left(\frac{y_j - \mathbf{x}_j\boldsymbol{\beta}}{\sigma}\right)^2 - \ln(\sqrt{2\pi}\sigma) & t_j = 0 \end{cases}$$

where  $\Phi(\cdot)$  is the cumulative distribution function of the standard normal distribution.

In the maximum likelihood estimation,  $\sigma$  and  $\rho$  are not directly estimated. Rather  $\ln\sigma$  and  $\operatorname{atanh}\rho$  are directly estimated, where

$$\operatorname{atanh}\rho = \frac{1}{2} \ln\left(\frac{1 + \rho}{1 - \rho}\right)$$

The standard error of  $\lambda = \rho\sigma$  is approximated through the delta method, which is given by

$$\text{Var}(\lambda) \approx \mathbf{D} \text{Var}\{(\text{atanh } \rho \ \ln\sigma)\} \mathbf{D}'$$

where  $\mathbf{D}$  is the Jacobian of  $\lambda$  with respect to  $\text{atanh } \rho$  and  $\ln\sigma$ .

Maddala (1983, 120–122) also derives the CF estimator as a two-step estimator. This estimator is implemented here. We will discuss it and then discuss the one-step CF estimator that is also implemented.

For the two-step estimator, probit estimates of the treatment equation

$$\Pr(t_j = 1 \mid \mathbf{w}_j) = \Phi(\mathbf{w}_j\boldsymbol{\gamma})$$

are obtained in the first stage. From these estimates, the hazard,  $h_j$ , for each observation  $j$  is computed as

$$h_j = \begin{cases} \phi(\mathbf{w}_j\hat{\boldsymbol{\gamma}})/\Phi(\mathbf{w}_j\hat{\boldsymbol{\gamma}}) & t_j = 1 \\ -\phi(\mathbf{w}_j\hat{\boldsymbol{\gamma}})/\{1 - \Phi(\mathbf{w}_j\hat{\boldsymbol{\gamma}})\} & t_j = 0 \end{cases}$$

where  $\phi$  is the standard normal density function. If

$$d_j = h_j(h_j + \mathbf{w}_j\hat{\boldsymbol{\gamma}})$$

then

$$\begin{aligned} E(y_j \mid t_j, \mathbf{x}_j, \mathbf{w}_j) &= \mathbf{x}_j\boldsymbol{\beta} + \delta t_j + \rho\sigma h_j \\ \text{Var}(y_j \mid t_j, \mathbf{x}_j, \mathbf{w}_j) &= \sigma^2 (1 - \rho^2 d_j) \end{aligned}$$

The two-step parameter estimates of  $\boldsymbol{\beta}$  and  $\delta$  are obtained by augmenting the regression equation with the hazard  $h$ . Thus the regressors become  $[\mathbf{x} \ t \ h]$ , and the additional parameter estimate  $\beta_h$  is obtained on the variable containing the hazard. A consistent estimate of the regression disturbance variance is obtained using the residuals from the augmented regression and the parameter estimate on the hazard

$$\hat{\sigma}^2 = \frac{\mathbf{e}'\mathbf{e} + \beta_h^2 \sum_{j=1}^N d_j}{N}$$

The two-step estimate of  $\rho$  is then

$$\hat{\rho} = \frac{\beta_h}{\hat{\sigma}}$$

To understand how the consistent estimates of the coefficient covariance matrix based on the augmented regression are derived, let  $\mathbf{A} = [\mathbf{x} \ t \ h]$  and  $\mathbf{D}$  be a square diagonal matrix of size  $N$  with  $(1 - \hat{\rho}^2 d_j)$  on the diagonal elements. The conventional VCE is

$$\mathbf{V}_{\text{two-step}} = \hat{\sigma}^2 (\mathbf{A}'\mathbf{A})^{-1} (\mathbf{A}'\mathbf{D}\mathbf{A} + \mathbf{Q}) (\mathbf{A}'\mathbf{A})^{-1}$$

where

$$\mathbf{Q} = \hat{\rho}^2 (\mathbf{A}'\mathbf{D}\mathbf{A}) \mathbf{V}_p (\mathbf{A}'\mathbf{D}\mathbf{A})$$

and  $\mathbf{V}_p$  is the variance–covariance estimate from the probit estimation of the treatment equation.

The one-step CF estimator is a GMM estimator with stacked moments. See Newey (1984) and Wooldridge (2010, sec. 14.2) for a description of this technique. Many econometric and statistical models can be expressed as conditions on the population moments. The parameter estimates produced by GMM estimators make the sample-moment conditions as true as possible given the data.

Under CF estimation, as in maximum likelihood estimation, we directly estimate  $\text{atanh } \rho$  and  $\ln \sigma$  rather than  $\rho$  and  $\sigma$ , so the parameter vector is

$$\boldsymbol{\theta} = (\boldsymbol{\beta}', \delta, \boldsymbol{\gamma}', \text{atanh } \rho, \ln \sigma)'$$

In this case, we have separate error functions for the treatment assignment

$$u_t(t_j, \mathbf{w}_j, \boldsymbol{\theta}) = \begin{cases} \phi(\mathbf{w}_j \boldsymbol{\gamma}) / \Phi(\mathbf{w}_j \boldsymbol{\gamma}) & t_j = 1 \\ -\phi(\mathbf{w}_j \boldsymbol{\gamma}) / \{1 - \Phi(\mathbf{w}_j \boldsymbol{\gamma})\} & t_j = 0 \end{cases}$$

for the outcome mean

$$u_m(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta}) = y_j - \mathbf{x}_j \boldsymbol{\beta} - \delta t_j - \rho \sigma u_{t,j}$$

and for the outcome variance

$$u_v(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta}) = u_{m,j}^2 - \sigma^2 [1 - \rho^2 \{u_{t,j}(u_{t,j} + \mathbf{w}_j \boldsymbol{\gamma})\}]$$

We calculate the hazard,  $h_j$ , prior to estimation from a probit regression of the treatment  $t_j$  on the treatment covariates  $\mathbf{w}_j$ . Let  $\tilde{\mathbf{z}}_j = (\mathbf{x}_j, t_j, h_j)$ . Now we define

$$\mathbf{Z}_j = \begin{bmatrix} \tilde{\mathbf{z}}_j & \mathbf{0} & 0 \\ 0 & \mathbf{w}_j & 0 \\ 0 & \mathbf{0} & 1 \end{bmatrix}$$

and

$$s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta}) = \mathbf{Z}_j' \begin{bmatrix} u_{m,j} \\ u_{t,j} \\ u_{v,j} \end{bmatrix}$$

The CF estimator  $\hat{\boldsymbol{\theta}}$  is the value of  $\boldsymbol{\theta}$  that satisfies the sample-moment conditions

$$\mathbf{0} = \frac{1}{N} \sum_i s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta})$$

The Huber/White/robust sandwich estimator is consistent for the VCE. See [Wooldridge \(2010, chap. 14\)](#), [Cameron and Trivedi \(2005, chap. 6\)](#), and [Newey and McFadden \(1994\)](#).

The formula is

$$\hat{\mathbf{V}} = (1/N) \overline{\mathbf{G}} \overline{\mathbf{S}} \overline{\mathbf{G}}'$$

where

$$\overline{\mathbf{G}} = \left\{ (1/N) \sum_i \frac{\partial s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta})}{\partial \hat{\boldsymbol{\theta}}} \right\}^{-1}$$

and

$$\overline{\mathbf{S}} = (1/N) \sum_i s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta}) s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta})'$$

The matrix  $\overline{\mathbf{G}}$  is not symmetric because our estimator comes from stacking the moment conditions instead of optimizing one objective function. The implication is that the robust formula should always be used because, even under correct specification, the nonsymmetric  $\overline{\mathbf{G}}$  and the symmetric  $\overline{\mathbf{S}}$  converge to different matrices.



## General potential-outcome model

Equation (1) can be generalized to a potential-outcome model with separate variance and correlation parameters for the control and treatment groups.

The generalized model is

$$\begin{aligned} y_{0j} &= \mathbf{x}_j \boldsymbol{\beta}_0 + \epsilon_{0j} \\ y_{1j} &= \mathbf{x}_j \boldsymbol{\beta}_1 + \epsilon_{1j} \\ t_j &= \begin{cases} 1, & \text{if } \mathbf{w}_j \boldsymbol{\gamma} + u_j > 0 \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

where  $y_{0j}$  is the outcome that person  $j$  obtains if person  $j$  selects treatment 0, and  $y_{1j}$  is the outcome that person  $j$  obtains if person  $j$  selects treatment 1. We never observe both  $y_{0j}$  and  $y_{1j}$ , only one or the other. We observe

$$y_j = t_j y_{1j} + (1 - t_j) y_{0j}$$

In this unconstrained model, the vector of error terms  $(\epsilon_{0j}, \epsilon_{1j}, u_j)'$  comes from a mean zero trivariate normal distribution with covariance matrix

$$\begin{bmatrix} \sigma_0^2 & \sigma_{01} & \sigma_0 \rho_0 \\ \sigma_{01} & \sigma_1^2 & \sigma_1 \rho_1 \\ \sigma_0 \rho_0 & \sigma_1 \rho_1 & 1 \end{bmatrix}$$

The likelihood function for this model is given in [Maddala \(1983, 224\)](#).

$$\ln f_j = \begin{cases} \ln \Phi \left\{ \frac{\mathbf{w}_j \boldsymbol{\gamma} + (y_j - \mathbf{x}_j \boldsymbol{\beta}_1) \rho_1 / \sigma_1}{\sqrt{1 - \rho_1^2}} \right\} - \frac{1}{2} \left( \frac{y_j - \mathbf{x}_j \boldsymbol{\beta}_1}{\sigma_1} \right)^2 - \ln(\sqrt{2\pi} \sigma_1), & t_j = 1 \\ \ln \Phi \left\{ \frac{-\mathbf{w}_j \boldsymbol{\gamma} - (y_j - \mathbf{x}_j \boldsymbol{\beta}_0) \rho_0 / \sigma_0}{\sqrt{1 - \rho_0^2}} \right\} - \frac{1}{2} \left( \frac{y_j - \mathbf{x}_j \boldsymbol{\beta}_0}{\sigma_0} \right)^2 - \ln(\sqrt{2\pi} \sigma_0), & t_j = 0 \end{cases}$$

$$\ln L = \sum_{j=1}^n w_j \ln f_j$$

where  $\Phi(\cdot)$  is the cumulative distribution function of the standard normal distribution, and  $w_j$  is an optional weight. The covariance between  $\epsilon_{0j}$  and  $\epsilon_{1j}$ ,  $\sigma_{01}$ , cannot be estimated because the potential outcomes  $y_{0j}$  and  $y_{1j}$  are never observed simultaneously.

As in the constrained model,  $\sigma_0$  and  $\sigma_1$  are not directly estimated in the maximum likelihood estimation; rather,  $\ln \sigma_0$  and  $\ln \sigma_1$  are estimated.

The parameters  $\rho_0$  and  $\rho_1$  are also not directly estimated; rather,  $\operatorname{atanh} \rho_0$  and  $\operatorname{atanh} \rho_1$  are directly estimated.

The new parameter vector is

$$\boldsymbol{\theta} = (\boldsymbol{\beta}'_0, \boldsymbol{\beta}'_1, \boldsymbol{\gamma}', \operatorname{atanh} \rho_0, \ln \sigma_0, \operatorname{atanh} \rho_1, \ln \sigma_1)'$$

The CF estimator for this potential-outcome model uses new error functions for the outcome mean

$$\begin{aligned} u_m(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta}) &= y_i - t_j (\mathbf{x}_j \boldsymbol{\beta}_1 + \rho_1 \sigma_1 u_{t,j}) \\ &\quad - (1 - t_j) (\mathbf{x}_j \boldsymbol{\beta}_0 + \rho_0 \sigma_0 u_{t,j}) \end{aligned}$$

and for the outcome variances

$$\begin{aligned} u_{v,0}(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta}) &= (1 - t_j) (u_{m,j}^2 - \sigma_0^2 [1 - \rho_0^2 \{u_{t,j}(u_{t,j} + \mathbf{w}_j\boldsymbol{\gamma})\}]) \\ u_{v,1}(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta}) &= t_j (u_{m,j}^2 - \sigma_1^2 [1 - \rho_1^2 \{u_{t,j}(u_{t,j} + \mathbf{w}_j\boldsymbol{\gamma})\}]) \end{aligned}$$

These error functions are derived based on the identities

$$\begin{aligned} E(y_j | t_j, \mathbf{x}_j, \mathbf{w}_j) &= t_j(\mathbf{x}_j\boldsymbol{\beta}_1 + \rho_1\sigma_1u_{t,j}) + (1 - t_j)(\mathbf{x}_j\boldsymbol{\beta}_0 + \rho_0\sigma_0u_{t,j}) \\ \text{Var}(y_j | t_j = 0, \mathbf{x}_j, \mathbf{w}_j) &= \sigma_0^2 [1 - \rho_0^2 \{u_{t,j}(u_{t,j} + \mathbf{w}_j\boldsymbol{\gamma})\}] \\ \text{Var}(y_j | t_j = 1, \mathbf{x}_j, \mathbf{w}_j) &= \sigma_1^2 [1 - \rho_1^2 \{u_{t,j}(u_{t,j} + \mathbf{w}_j\boldsymbol{\gamma})\}] \end{aligned}$$

We calculate the hazard,  $h_j$ , prior to estimation from a probit regression of the treatment,  $t_j$ , on the treatment covariates,  $\mathbf{w}_j$ . Let  $\tilde{\mathbf{z}}_j = \{\mathbf{x}_j, t_j h_j, (1 - t_j) h_j\}$ . Now we define

$$\mathbf{Z}_j = \begin{bmatrix} \tilde{\mathbf{z}}_j & \mathbf{0} & 0 & 0 \\ 0 & \mathbf{w}_j & 0 & 0 \\ 0 & \mathbf{0} & 1 & 0 \\ 0 & \mathbf{0} & 0 & 1 \end{bmatrix}$$

and

$$s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta}) = \mathbf{Z}'_j \begin{bmatrix} u_{m,j} \\ u_{t,j} \\ u_{v,0,j} \\ u_{v,1,j} \end{bmatrix}$$

The CF estimator  $\hat{\boldsymbol{\theta}}$  is the value of  $\boldsymbol{\theta}$  that satisfies the sample-moment conditions

$$\mathbf{0} = \frac{1}{N} \sum_i s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta})$$

The Huber/White/robust sandwich estimator is consistent for the VCE. See [Wooldridge \(2010, chap. 14\)](#), [Cameron and Trivedi \(2005, chap. 6\)](#), and [Newey and McFadden \(1994\)](#).

The formula is

$$\hat{\mathbf{V}} = (1/N) \overline{\mathbf{G}} \overline{\mathbf{S}} \overline{\mathbf{G}}'$$

where

$$\overline{\mathbf{G}} = \left\{ (1/N) \sum_i \frac{\partial s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right\}^{-1}$$

and

$$\overline{\mathbf{S}} = (1/N) \sum_i s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta}) s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta})'$$

The matrix  $\overline{\mathbf{G}}$  is not symmetric because our estimator comes from stacking the moment conditions instead of optimizing one objective function. The implication is that the robust formula should always be used because, even under correct specification, the nonsymmetric  $\overline{\mathbf{G}}$  and the symmetric  $\overline{\mathbf{S}}$  converge to different matrices.

## Average treatment effect

The ATE is the average difference of the treated potential outcomes and the control potential outcomes.

By the law of iterated expectations, the ATE is

$$\begin{aligned} E(y_{1j} - y_{0j}) &= E\{E(y_{1j} - y_{0j} | \mathbf{x}_j, \epsilon_{0j}, \epsilon_{1j})\} \\ &= E(\mathbf{x}_j \beta_1 + \epsilon_1 - \mathbf{x}_j \beta_0 - \epsilon_0) \\ &= E\{\mathbf{x}_j(\beta_1 - \beta_0)\} \end{aligned}$$

This expectation can be estimated as a predictive margin when  $\mathbf{x}_j(\beta_1 - \beta_0)$  varies in  $\mathbf{x}_j$ . Otherwise, the ATE is estimated as the coefficient of  $t_j$  in the model.

## Average treatment effect on the treated

The ATE is the average difference of the treated potential outcomes and the control potential outcomes on the treated population.

The conditional means of the potential outcomes  $y_{tj}$ ,  $t \in (0, 1)$  for exogenous covariates  $\mathbf{x}_j$  and treatment covariates  $\mathbf{w}_j$  at treatment  $t_j = 1$  are

$$E(y_{tj} | \mathbf{x}_j, \mathbf{w}_j, t_j = 1) = \mathbf{x}_j \beta_t + \rho_t \sigma_t \phi(\mathbf{w}_j \boldsymbol{\gamma}) / \Phi(\mathbf{w}_j \boldsymbol{\gamma})$$

By the law of iterated expectations, the ATET is

$$\begin{aligned} E(y_{1j} - y_{0j} | t_j = 1) &= E\{E(y_{1j} - y_{0j} | \mathbf{x}_j, \mathbf{w}_j, t_j = 1)\} \\ &= E\{\mathbf{x}_j(\beta_1 - \beta_0) + (\rho_1 \sigma_1 - \rho_0 \sigma_0) \phi(\mathbf{w}_j \boldsymbol{\gamma}) / \Phi(\mathbf{w}_j \boldsymbol{\gamma}) | t_j = 1\} \end{aligned}$$

This expectation can be estimated as a predictive margin on the treated population when  $\mathbf{x}_j(\beta_1 - \beta_0)$  varies in  $\mathbf{x}_j$  or when the variance and correlation parameters differ by treatment group. Otherwise, the ATET is estimated as the coefficient of  $t_j$  in the model.

## References

- Barnow, B. S., G. G. Cain, and A. S. Goldberger. 1981. Issues in the analysis of selectivity bias. In Vol. 5 of *Evaluation Studies Review Annual*, ed. E. W. Stromsdorfer and G. Farkas, 123–126. Beverly Hills: SAGE.
- Brave, S., and T. Walstrum. 2014. Estimating marginal treatment effects using parametric and semiparametric methods. *Stata Journal* 14: 191–217.
- Brown, G. K., and T. Mergoupis. 2011. Treatment interactions with nonexperimental data in Stata. *Stata Journal* 11: 545–555.
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cerulli, G. 2015a. *Econometric Evaluation of Socio-Economic Programs: Theory and Applications*. Berlin: Springer.
- . 2015b. `ctreatreg`: Command for fitting dose-response models under exogenous and endogenous treatment. *Stata Journal* 15: 1019–1045.
- Greene, W. H. 2000. *Econometric Analysis*. 4th ed. Upper Saddle River, NJ: Prentice Hall.
- . 2012. *Econometric Analysis*. 7th ed. Upper Saddle River, NJ: Prentice Hall.

- Heckman, J. 1976. The common structure of statistical models of truncation, sample selection and limited dependent variables and a simple estimator for such models. *Annals of Economic and Social Measurement* 5: 475–492.
- . 1978. Dummy endogenous variables in a simultaneous equation system. *Econometrica* 46: 931–959. <https://doi.org/10.2307/1909757>.
- Maddala, G. S. 1983. *Limited-Dependent and Qualitative Variables in Econometrics*. Cambridge: Cambridge University Press.
- Nannicini, T. 2007. Simulation-based sensitivity analysis for matching estimators. *Stata Journal* 7: 334–350.
- Newey, W. K. 1984. A method of moments interpretation of sequential estimators. *Economics Letters* 14: 201–206. [https://doi.org/10.1016/0165-1765\(84\)90083-1](https://doi.org/10.1016/0165-1765(84)90083-1).
- Newey, W. K., and D. L. McFadden. 1994. Large sample estimation and hypothesis testing. In Vol. 4 of *Handbook of Econometrics*, ed. R. F. Engle and D. L. McFadden, 2111–2245. Amsterdam: Elsevier. [https://doi.org/10.1016/S1573-4412\(05\)80005-4](https://doi.org/10.1016/S1573-4412(05)80005-4).
- Nichols, A. 2007. Causal inference with observational data. *Stata Journal* 7: 507–541.
- Vella, F., and M. Verbeek. 1998. Whose wages do unions raise? A dynamic model of unionism and wage rate determination for young men. *Journal of Applied Econometrics* 13: 163–183. [https://doi.org/10.1002/\(SICI\)1099-1255\(199803/04\)13:2<163::AID-JAE460>3.0.CO;2-Y](https://doi.org/10.1002/(SICI)1099-1255(199803/04)13:2<163::AID-JAE460>3.0.CO;2-Y).
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

## Also see

- [TE] **etregress postestimation** — Postestimation tools for etregress
- [TE] **etpoisson** — Poisson regression with endogenous treatment effects
- [ERM] **eregress** — Extended linear regression
- [R] **heckman** — Heckman selection model
- [R] **probit** — Probit regression
- [R] **regress** — Linear regression
- [SVY] **svy estimation** — Estimation commands for survey data
- [U] **20 Estimation and postestimation commands**

[Postestimation commands](#)   
 [predict](#)   
 [margins](#)   
 [Remarks and examples](#)  
 Also see

## Postestimation commands

The following postestimation commands are available after `etregress`:

Command	Description
<code>contrast</code>	contrasts and ANOVA-style joint tests of estimates
* <code>estat ic</code>	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance-covariance matrix of the estimators (VCE)
<code>estat (svy)</code>	postestimation statistics for survey data
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
* <code>hausman</code>	Hausman's specification test
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of coefficients
* <code>lrtest</code>	likelihood-ratio test
<code>margins</code>	marginal means, predictive margins, marginal effects, and average marginal effects
<code>marginsplot</code>	graph the results from margins (profile plots, interaction plots, etc.)
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
<code>predict</code>	conditional treatment effects, linear predictions and their SEs, etc.
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>pwcompare</code>	pairwise comparisons of estimates
* <code>suest</code>	seemingly unrelated estimation
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

\*`estat ic`, `lrtest`, and `suest` are not appropriate after `etregress`, `twostep` or `etregress`, `cfunction`.  
`hausman` and `lrtest` are not appropriate with `svy` estimation results.

# predict

## Description for predict

`predict` creates a new variable containing predictions such as linear predictions, conditional treatment effects, standard errors, expected values, and probabilities.

## Menu for predict

Statistics > Postestimation

## Syntax for predict

After *ML*, *twostep*, or *cfunction*

```
predict [type] newvar [if] [in] [, statistic]
```

After *ML* or *cfunction*

```
predict [type] stub* [if] [in], scores
```

<i>statistic</i>	Description
------------------	-------------

<i>statistic</i>	Description
Main	
<code>xb</code>	linear prediction; the default
<code>cte</code>	conditional treatment effect at treatment level
<code>stdp</code>	standard error of the prediction
<code>stdf</code>	standard error of the forecast
<code>ycntrt</code>	$E(y_j \mid \text{treatment} = 1)$
<code>ycntrt</code>	$E(y_j \mid \text{treatment} = 0)$
<code>ptrt</code>	$\Pr(\text{treatment} = 1)$
<code>xbtrt</code>	linear prediction for treatment equation
<code>stdp<sup>trt</sup></code>	standard error of the linear prediction for treatment equation

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

`stdf` is not allowed with `svy` estimation results.

## Options for predict

Main

`xb`, the default, calculates the linear prediction,  $\mathbf{x}_j\mathbf{b}$ .

`cte` calculates the treatment effect, the difference of potential-outcome means, conditioned on treatment level.

`stdp` calculates the standard error of the prediction, which can be thought of as the standard error of the predicted expected value or mean for the observation's covariate pattern. The standard error of the prediction is also referred to as the standard error of the fitted value.

`stdf` calculates the standard error of the forecast, which is the standard error of the point prediction for one observation. It is commonly referred to as the standard error of the future or forecast value. By construction, the standard errors produced by `stdf` are always larger than those produced by `stdp`; see *Methods and formulas* in [R] [regress postestimation](#).

`yctr` calculates the expected value of the dependent variable conditional on the presence of the treatment:  $E(y_j \mid \text{treatment} = 1)$ .

`ycntr` calculates the expected value of the dependent variable conditional on the absence of the treatment:  $E(y_j \mid \text{treatment} = 0)$ .

`ptr` calculates the probability of the presence of the treatment:

$$\Pr(\text{treatment} = 1) = \Pr(\mathbf{w}_j\boldsymbol{\gamma} + u_j > 0).$$

`xbtr` calculates the linear prediction for the treatment equation.

`stdptr` calculates the standard error of the linear prediction for the treatment equation.

`scores`, not available with `twostep`, calculates equation-level score variables.

The first new variable will contain  $\partial \ln L / \partial (\mathbf{x}_j\boldsymbol{\beta})$ .

The second new variable will contain  $\partial \ln L / \partial (\mathbf{w}_j\boldsymbol{\gamma})$ .

Under the constrained model, the third new variable will contain  $\partial \ln L / \partial \text{atanh } \rho$ .

Under the constrained model, the fourth new variable will contain  $\partial \ln L / \partial \ln \sigma$ .

Under the general potential-outcome model, the third new variable will contain  $\partial \ln L / \partial \text{atanh } \rho_0$ .

Under the general potential-outcome model, the fourth new variable will contain  $\partial \ln L / \partial \ln \sigma_0$ .

Under the general potential-outcome model, the fifth new variable will contain  $\partial \ln L / \partial \text{atanh } \rho_1$ .

Under the general potential-outcome model, the sixth new variable will contain  $\partial \ln L / \partial \ln \sigma_1$ .

# margins

## Description for margins

`margins` estimates margins of response for linear predictions, conditional treatment effects, expected values, and probabilities.

## Menu for margins

Statistics > Postestimation

## Syntax for margins

```
margins [marginlist] [, options]
```

```
margins [marginlist] , predict(statistic ...) [predict(statistic ...) ...] [options]
```

### Maximum likelihood and control-function estimation results

<i>statistic</i>	Description
<code>xb</code>	linear prediction; the default
<code>cte</code>	conditional treatment effect at treatment level
<code>ycntrt</code>	$E(y_j \mid \text{treatment} = 1)$
<code>ycntrt</code>	$E(y_j \mid \text{treatment} = 0)$
<code>ptrt</code>	$\Pr(\text{treatment} = 1)$
<code>xbtrt</code>	linear prediction for treatment equation
<code>stdp</code>	not allowed with <code>margins</code>
<code>stdf</code>	not allowed with <code>margins</code>
<code>stdp<u>trt</u></code>	not allowed with <code>margins</code>

### Two-step estimation results

<i>statistic</i>	Description
<code>xb</code>	linear prediction; the default
<code>ptrt</code>	$\Pr(\text{treatment} = 1)$
<code>xbtrt</code>	linear prediction for treatment equation
<code>cte</code>	not allowed with <code>margins</code>
<code>ycntrt</code>	not allowed with <code>margins</code>
<code>ycntrt</code>	not allowed with <code>margins</code>
<code>stdp</code>	not allowed with <code>margins</code>
<code>stdf</code>	not allowed with <code>margins</code>
<code>stdp<u>trt</u></code>	not allowed with <code>margins</code>

Statistics not allowed with `margins` are functions of stochastic quantities other than  $e(b)$ .

For the full syntax, see [\[R\] margins](#).



## Remarks and examples

The average treatment effect (ATE) and the average treatment effect on the treated (ATET) are the parameters most frequently estimated by postestimation techniques after `etregress`.

When there are no interactions between the treatment variable and the outcome covariates in the constrained model, `etregress` directly estimates the ATE and the ATET; see [example 1](#) of [\[TE\] etregress](#).

When there are no interactions between the treatment variable and the outcome covariates in the general potential-outcome model, `etregress` directly estimates the ATE; see [example 2](#) of [\[TE\] etregress](#).

When there are interactions between the treatment variable and the outcome covariates, you can use `margins` after `etregress` to estimate the ATE. See [example 3](#) and [example 4](#) of [\[TE\] etregress](#) for examples of ATE estimation.

When there are interactions between the treatment variable and the outcome covariates in the constrained model, you can use `margins` after `etregress` to estimate the ATET. See [example 5](#) of [\[TE\] etregress](#) for an example of ATET estimation in the constrained model.

In the general potential-outcome model, you can use `margins` after `etregress` to estimate the ATET. See [example 6](#) of [\[TE\] etregress](#) for an example of ATET estimation in the general potential-outcome model.

## Also see

[\[TE\] etregress](#) — Linear regression with endogenous treatment effects

[\[U\] 20 Estimation and postestimation commands](#)

# Title

**stteffects** — Treatment-effects estimation for observational survival-time data

[Description](#)   [Syntax](#)   [Also see](#)

## Description

`stteffects` estimates average treatment effects, average treatment effects on the treated, and potential-outcome means using observational survival-time data. The available estimators are regression adjustment, inverse-probability weighting, and more efficient methods that combine regression adjustment and inverse-probability weighting.

For a brief description and example of each estimator, see [Remarks and examples](#) in [TE] [stteffects intro](#).

## Syntax

```
stteffects subcommand ... [ , options ]
```

<i>subcommand</i>	Description
<a href="#">ra</a>	regression adjustment
<a href="#">ipw</a>	inverse-probability weighting
<a href="#">ipwra</a>	inverse-probability-weighted regression adjustment
<a href="#">wra</a>	weighted regression adjustment

## Also see

[TE] [stteffects intro](#) — Introduction to treatment effects for observational survival-time data

# Title

**stteffects intro** — Introduction to treatment effects for observational survival-time data

[Description](#)  
[Also see](#)

[Remarks and examples](#)

[Acknowledgments](#)

[References](#)

## Description

This entry provides an overview of the treatment-effects estimators that use observational survival-time data and are implemented in `stteffects`. It also provides an overview of the potential-outcomes framework and its application to survival-time data and to the interpretation of the treatment-effects parameters estimated.

The `stteffects` command estimates average treatment effects (ATEs), average treatment effects on the treated (ATETs), and potential-outcome means (POMs). Each of these effect parameters is discussed in this entry. `stteffects` implements a variety of estimators for the ATE, ATET, and POM. The treatment effects can be estimated using regression adjustment (RA), inverse-probability weights (IPW), inverse-probability-weighted regression adjustment (IPWRA), and weighted regression adjustment (WRA). This entry also provides some intuition for the estimators and discusses the trade-offs between them.

## Remarks and examples

Remarks are presented under the following headings:

*[Introduction](#)*

*[A quick tour of the estimators](#)*

*[Regression adjustment](#)*

*[Inverse-probability weighting](#)*

*[Combinations of RA and IPW](#)*

*[Weighted regression adjustment](#)*

*[Average treatment effect on the treated](#)*

*[Comparison of treatment-effects estimators](#)*

*[Assumptions and trade-offs](#)*

*[The conditional independence assumption](#)*

*[The sufficient overlap assumption](#)*

*[The correct adjustment for censoring assumption](#)*

*[Assumptions for the ATET](#)*

*[Specification diagnostics and tests](#)*

*[Multivalued treatments](#)*

## Introduction

The `stteffects` command estimates treatment effects using observational survival-time data.

For some intuition about the methods implemented in the `stteffects` command, consider the following question: Does smoking decrease the time to a second heart attack in the population of women aged 45–55 who have had one heart attack? Three aspects of this question stand out.

1. For ethical reasons, these data will be observational.
2. This question is about the time to an event, and such data are commonly known as survival-time data or time-to-event data. These data are nonnegative and, frequently, right-censored.
3. Many researchers and practitioners want an effect estimate in easy-to-understand units of time.

`Aspect 1` is one of the most common reasons for using observational data, and `aspect 2` focuses interest on survival-time data.

We are most concerned with `aspect 3` because it helps us define and understand the effect of interest. In particular, we would like to know the average change in time to a second heart attack that would occur in the population if all women smoked instead of if no women smoked. This effect is an ATE.

We must solve a missing-data problem to estimate the ATE. The ATE is the population average of the contrast in outcomes when everyone gets the treatment and when no one gets the treatment. Formally, we write this as

$$\text{ATE} = E(t_1 - t_0)$$

where  $t_1$  is the survival time when a subject gets the treatment and  $t_0$  is the survival time when a subject does not get the treatment. For each treatment level, there is a potential outcome that would be observed if a subject received that treatment level:  $t_1$  is the potential outcome that would occur if someone gets the treatment and  $t_0$  is the potential outcome that would occur if someone does not get the treatment. The missing-data problem arises because each subject receives only one treatment level, and so we observe only one of the two potential outcomes.

Much of the survival-time literature uses a hazard ratio as the effect of interest. The ATE has three advantages over the hazard ratio as an effect measure.

1. The ATE measures the effect in the same time units as the outcome instead of in relative conditional probabilities.
2. The ATE is much easier to explain to nontechnical audiences.
3. The models used to estimate the ATE can be much more flexible. Hazard ratios are useful for population effects when they are constant, which occurs when the treatment enters linearly and the distribution of the outcome has a proportional-hazards form. Neither linearity in treatment nor proportional-hazards form is required for the ATE, and neither is imposed on the models fit by the estimators implemented in `stteffects`.

The estimators implemented in `stteffects` use the common missing-data techniques of regression modeling, weighting, and combinations thereof to account for data lost to censoring and to unobserved potential outcomes.

Here we note only a few contributions and entry points to the vast literature on estimating ATEs. The use of potential outcomes to define treatment effects has proved extraordinarily useful; see [Holland \(1986\)](#), [Rubin \(1974\)](#), and [Heckman \(1997\)](#). [Cameron and Trivedi \(2005, chap. 25\)](#), [Wooldridge \(2010, chap. 21\)](#), and [Vittinghoff et al. \(2012, chap. 9\)](#) provide excellent general introductions to estimating ATEs.

## □ Technical note

Left-truncation would be another type of missing data. The estimators implemented in `stteffects` do not adjust for left-truncation, so `stteffects` cannot be used with delayed-entry data.

`stteffects` cannot be used with time-varying covariates or multiple-record data because these add a repeated-measure structure that significantly complicates the estimation problem.

□

## A quick tour of the estimators

The `stteffects` command implements five estimators of treatment effects. We introduce each one by showing the basic syntax used to apply it to a common example dataset. See each command's entry for detailed information.

We have some fictional data on the time to a second heart attack among women aged 45–55 years. The treatment, smoking, is stored in the 0/1 indicator `smoke`. These data also contain each woman's age at the time of her first heart attack (`age`), and indices of her exercise level (`exercise`), diet quality (`diet`), and education attainment (`education`) prior to her first heart attack.

Like `streg` and other survival-time commands, `stteffects` uses the outcome variable and the failure indicator computed by `stset`. In this dataset, `atime` is the observed time in years to the second heart attack, and `fail` is the 0/1 indicator that a second heart attack was observed and recorded in `atime`. (When `fail` is 1, `atime` records the time to the second attack; when `fail` is 0, `atime` records a censored observation of the time to the second attack.)

We begin our examples by first reading in the data and then specifying the raw outcome and failure variables to `stset`.

```
. use https://www.stata-press.com/data/r17/sheart
(Time to second heart attack (fictional))
. stset atime, failure(fail)
Survival-time data settings
      Failure event: fail!=0 & fail<.
Observed time interval: (0, atime]
Exit on or before: failure
```

---

```
      2,000 total observations
         0 exclusions
```

---

```
      2,000 observations remaining, representing
      1,208 failures in single-record/single-failure data
3,795.226 total analysis time at risk and under observation
                At risk from t =           0
                Earliest observed entry t =           0
                Last observed exit t = 34.17743
```

The output indicates that 1,208 of the 2,000 observations record actual time to a second heart attack. The remaining observations were censored. Now that we have `stset` the data, we can use `stteffects`.

## Regression adjustment

Regression modeling of the outcome variable is a venerable approach to solving the missing-data problem in treatment-effects estimation. Known as the regression-adjustment (RA) estimator, this method uses averages of predicted outcomes to estimate the ATE. If the outcome model is well specified, this approach is surprisingly robust.

## ▷ Example 1: RA estimation

We now use `stteffects ra` to estimate the ATE by RA. We model the outcome as a function of age, exercise, diet, and education, and we specify that `smoke` is the treatment variable.

```
. stteffects ra (age exercise diet education) (smoke)
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:   EE criterion = 1.006e-14
Iteration 1:   EE criterion = 2.305e-25
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : regression adjustment
Outcome model  : Weibull
Treatment model: none
Censoring model: none
```

		Robust				
	_t	Coefficient	std. err.	z	P> z	[95% conf. interval]
ATE	smoke (Smoker vs Nonsmoker)	-1.956657	.3331787	-5.87	0.000	-2.609676 -1.303639
POmean	smoke Nonsmoker	4.243974	.2620538	16.20	0.000	3.730358 4.75759

When all women in the population smoke, the average time to a second heart attack is estimated to be 1.96 years less than when no women smoke. The estimated average time to a second heart attack when no women smoke is 4.24 years.

The output reports that a Weibull model was used for the outcome. The other outcome models available are exponential, gamma, and log normal. See [example 2 in \[TE\] stteffects ra](#) for an application of the gamma parameterization to this model.

The ratio of the ATE to control-level POM measures the importance of the effect. In this example, when all women smoke, the time to a second heart attack falls by an estimated 46% relative to the case in which none of them smoke. See [example 3 in \[TE\] stteffects ra](#) for an example that uses `nlcom` to compute a point estimate and a confidence interval for this ratio.

◀

Unlike the IPW estimator discussed in the next section, RA does not model treatment assignment or the censoring process. Treatment assignment is handled by fitting separate models for each treatment level and averaging the predicted outcomes. As is standard in the survival-time literature, the censoring term in the log-likelihood function accounts for censoring; see [Kalbfleisch and Prentice \(2002, chap. 3\)](#), [Cameron and Trivedi \(2005, chap. 17\)](#), [Cleves, Gould, and Marchenko \(2016, chap. 13\)](#), and [Wooldridge \(2010, chap. 22\)](#).

See [\[TE\] stteffects ra](#) for further discussion of this command and the RA estimator.

## Inverse-probability weighting

Sometimes researchers are more comfortable modeling treatment assignment than the outcome. Inverse-probability-weighted (IPW) estimators use weighted averages of the observed outcome to estimate the POMS and the ATE. The weights correct for the missing data. When there is no censoring, the missing potential outcome is the only missing data, and the weights are constructed from a model of treatment assignment. When the data may be censored, the weights must control for censoring and the missing potential outcome. In this case, IPW estimators construct the weights from two models, one for the censoring time and one for treatment assignment.

### ► Example 2: IPW estimation

Here we use `stteffects ipw` to estimate the effect of smoking on the time to a second heart attack. The model of assignment to the treatment `smoke` depends on `age`, `exercise`, `diet`, and `education`. The time-to-censoring model also depends on `age`, `exercise`, `diet`, and `education`.

```
. stteffects ipw (smoke age exercise diet education)
> (age exercise diet education)

      Failure _d: fail
      Analysis time _t: atime

Iteration 0:  EE criterion = 2.042e-18
Iteration 1:  EE criterion = 3.283e-31

Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: logit
Censoring model: Weibull
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE smoke (Smoker vs Nonsmoker)	-2.187297	.6319837	-3.46	0.001	-3.425962	-.9486314
POmean smoke Nonsmoker	4.225331	.517501	8.16	0.000	3.211047	5.239614

When all women in the population smoke, the average time to a second heart attack is estimated to be 2.19 years less than when no women smoke. The estimated average time to a second heart attack when no women smoke is 4.23 years. When all women smoke, the average time to a second heart attack falls by an estimated 52% relative to the case when no women smoke.

The estimates have changed; however, the interpretation is the same as for the RA estimator because the IPW and RA estimators are estimating the same population effects. Under correct model specification, the estimates will differ in finite samples, but the size of these differences will decrease as the sample size gets larger. For the case at hand, the estimated ATE and control-level POM are roughly similar to those produced by the RA estimator using the Weibull model for the outcome.

Recall that IPW estimators are weighted averages of observed outcomes and that the weights control for the missing outcomes. Weights in survival-time data have two components: one for the missing potential outcome and one for data lost to censoring. We used a logit model for treatment assignment, so the component of the weights that controls for the missing potential outcome comes from the

estimated logit parameters. We used a Weibull model for the time to censoring, so the component of the weights that controls for data lost to censoring comes from the estimated Weibull parameters. ↵

Using weighting from an estimated treatment-assignment model to control for the missing potential outcome is standard in the treatment-effects literature; for example, see [TE] **teffects intro advanced**, Wooldridge (2010, chap. 21), Vittinghoff et al. (2012, chap. 9), Hirano, Imbens, and Ridder (2003), Cattaneo (2010), and Cattaneo, Drukker, and Holland (2013). Modeling the time to censoring is specific to the survival-time treatment-effects literature; see Bai, Tsiatis, and O'Brien (2013) and Robins and Rotnitzky (2006). See *Methods and formulas* in [TE] **stteffects ipwra** for more details.

See [TE] **stteffects ipw** for further discussion of this command and the IPW estimator.

## Combinations of RA and IPW

More efficient estimators are obtained by combining IPW and RA, due to Wooldridge (2007) and Wooldridge (2010, chap. 21) and denoted by IPWRA. Unlike the estimators discussed in Wooldridge (2010, chap. 21), both the treatment and the outcome models must be correctly specified to estimate the ATE.

The IPWRA estimator uses estimated weights that control for missing data to obtain missingness-adjusted regression coefficients that are used to compute averages of predicted outcomes to estimate the POMs. The estimated ATE is a contrast of the estimated POMs. These weights always involve a model for treatment assignment. You choose whether to account for censoring by including a term in the log-likelihood function or whether to use weights that also account for the data lost to censoring.

## ▷ Example 3: Likelihood-adjusted-censoring IPWRA estimation

We model the outcome (time to a second heart attack) as a function of age, exercise, diet, and education. We model assignment to the treatment smoke as a function of the same covariates.

```
. stteffects ipwra (age exercise diet education)
> (smoke age exercise diet education)

      Failure _d: fail
      Analysis time _t: atime
Iteration 0:   EE criterion = 6.476e-15
Iteration 1:   EE criterion = 1.116e-26
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : IPW regression adjustment
Outcome model  : Weibull
Treatment model: logit
Censoring model: none
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE smoke (Smoker vs Nonsmoker)	-1.592494	.4872777	-3.27	0.001	-2.54754	-.637447
POmean smoke Nonsmoker	4.214523	.2600165	16.21	0.000	3.7049	4.724146

The estimated ATE of  $-1.59$  and control-level POM of  $4.21$  are similar to the reported values of  $-1.96$  and  $4.24$  in [example 1](#).



We did not specify a model for the time to censoring, so censoring is handled by including a term in the log-likelihood function in the Weibull outcome model. We denote this likelihood-adjusted-censoring (LAC) version of the IPWRA estimator by LAC-IPWRA.



► Example 4: Weighted-adjusted-censoring IPWRA estimation

Instead of including a term in the log-likelihood function, the weighted-adjusted-censoring IPWRA (WAC-IPWRA) estimator uses estimated weights to adjust for censoring. We model the time to a second heart attack as a function of `age`, `exercise`, `diet`, and `education`; we model assignment to the treatment `smoke` as a function of the same covariates; and we model the time to censoring as a function of `age`, `exercise`, and `diet`.

```
. stteffects ipwra (age exercise diet education)
> (smoke age exercise diet education) (age exercise diet)

      Failure _d: fail
      Analysis time _t: atime
Iteration 0:   EE criterion = 2.797e-13
Iteration 1:   EE criterion = 2.032e-25
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : IPW regression adjustment
Outcome model  : Weibull
Treatment model: logit
Censoring model: Weibull
```

_t		Robust		z	P> z	[95% conf. interval]	
		Coefficient	std. err.				
ATE							
smoke							
(Smoker							
vs							
Nonsmoker)		-2.037944	.6032549	-3.38	0.001	-3.220302	-.855586
POMean							
smoke							
Nonsmoker		4.14284	.4811052	8.61	0.000	3.199891	5.085789

The estimated ATE of  $-2.04$  and control-level POM of  $4.14$  are similar to the reported values of  $-1.96$  and  $4.24$  in [example 1](#).

The weights for censoring are constructed from the estimated parameters because we specified a time-to-censoring model.



Under correct specification, both versions of the IPWRA estimator estimate the same ATE and control-level POM as estimated by RA and IPW.

The addition of the time-to-censoring model makes the WAC-IPWRA somewhat less robust than the LAC-IPWRA estimator. Weighting methods to control for censoring also place more restrictive assumptions on the censoring process. For example, the censoring time must be random, otherwise it would be impossible to construct the weights. In [Assumptions and trade-offs](#) below, we discuss the trade-offs among the estimators and the assumptions that each requires. For the moment, we note that we believe the LAC-IPWRA estimator is more robust than the WAC-IPWRA estimator.

See [\[TE\] stteffects ipwra](#) for further discussion of this command and the IPWRA estimator.

## Weighted regression adjustment

When estimating the parameters of an outcome model, the weighted regression-adjustment (WRA) estimator uses weights instead of a term in the log-likelihood function to adjust for censoring. These weights are constructed from a model for the censoring process. The estimated parameters are subsequently used to compute averages of predicted outcomes that estimate the POMs. A contrast of the estimated POMs estimates the ATE.

### ▷ Example 5: WRA estimation

We model the time to a second heart attack as a function of age, exercise, diet, and education; we specify that smoke is the treatment; and we model the time to censoring as a function of age, exercise, and diet.

```
. stteffects wra (age exercise diet education) (smoke) (age exercise diet)
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 7.116e-15
Iteration 1:  EE criterion = 5.859e-27
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : weighted regression adjustment
Outcome model  : Weibull
Treatment model: none
Censoring model: Weibull
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE smoke (Smoker vs Nonsmoker)	-2.152014	.4986005	-4.32	0.000	-3.129253	-1.174775
POMean smoke Nonsmoker	4.079273	.4379517	9.31	0.000	3.220903	4.937642

The estimated ATE of  $-2.15$  and control-level POM of  $4.08$  are similar to the reported values of  $-1.96$  and  $4.24$  in [example 1](#). Like the other estimators discussed, the WRA estimators estimate the same effect parameters as the RA estimator, so the interpretation is the same.

◀

In many survival-time applications, using weights to adjust for censoring is probably less robust than just including a term in the log-likelihood function for the outcome model. The model used to construct the weights is just as complicated as the outcome model, and including the term in the log-likelihood function places fewer restrictions on the censoring process, as discussed in [The correct adjustment for censoring assumption](#) below.

See [\[TE\] stteffects wra](#) for further discussion of this command and the WRA estimator.

## Average treatment effect on the treated

Intuitively, the average treatment effect on the treated (ATET) is the effect in a well-defined, at-risk subpopulation. Sometimes the subpopulation that gets the treatment defines such an at-risk subpopulation. For example, we may want to know the average change in time to a second heart attack among female smokers aged 45–55 who have had a heart attack if they all became nonsmokers. This effect is the ATET.

Below, we use `stteffects ra` to estimate the ATET by RA.

```
. stteffects ra (age exercise diet education) (smoke), atet
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:   EE criterion = 1.006e-14
Iteration 1:   EE criterion = 2.970e-26
Survival treatment-effects estimation          Number of obs      =      2,000
Estimator      : regression adjustment
Outcome model  : Weibull
Treatment model: none
Censoring model: none
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATET						
smoke (Smoker vs Nonsmoker)	-1.527476	.2489203	-6.14	0.000	-2.015351	-1.039602
POmean						
smoke Nonsmoker	3.436937	.2217808	15.50	0.000	3.002255	3.87162

Now, all effects are calculated only for the subpopulation of women aged 45–55 years who smoke after their first heart attack. If no women in the subpopulation were to smoke, the average time to a second heart attack would be 3.44 years. When all women in the subpopulation smoke (the observed behavior), the average time to a second heart attack is estimated to be 1.53 years less than if no women in the subpopulation had smoked. In other words, if we could somehow turn all smokers in the subpopulation into nonsmokers, the average time to a second heart attack would be 3.44 years instead of 1.91 years ( $3.44 - 1.53 = 1.91$ ).

These point estimates are a little different than those for the ATE and the control-level POM in the full population of women aged 45–55 years who have had one heart attack. The difference indicates that this particular health cost of smoking may be smaller among women who choose to smoke than in the full population.

## Comparison of treatment-effects estimators

We can classify the estimators implemented in `stteffects` into five categories: 1) estimators based on a model for the outcome variable; 2) estimators based on models for the treatment assignment and the censoring time; 3) estimators based on models for the outcome variable and the treatment assignment; 4) estimators based on models for the outcome variable, the treatment assignment, and the censoring time; and 5) estimators based on models for the outcome variable and the censoring time.

Because there are several categories of estimators, the user must decide whether to model the outcome, the probability of treatment, the time to censoring, or some combination thereof.

Each category of estimator contains a variety of choices about the functional forms for the models.

We now provide some intuition behind each category of estimator and discuss the relationships.

1. When modeling only the outcome, separate outcome models for each treatment level account for treatment assignment, and censoring is adjusted for in the log-likelihood function. This approach is used in the RA estimators.
2. Some researchers would rather avoid modeling the outcome. Some estimators use weighted averages of the observed outcome to estimate the effect. When estimating treatment effects from observational survival-time data, the weights used must account for treatment assignment and censoring. Models for treatment assignment and time to censoring are used to construct the weights. This approach is used in the IPW estimators.
3. When seeking a more efficient estimator, it is natural to model both the outcome and the treatment and to adjust for censoring in the outcome model. This approach is used in the LAC-IPWRA estimators.
4. When seeking a more efficient estimator, another natural approach is to model both the outcome and the treatment and to adjust for censoring by weights that come from a time-to-censoring model. This approach is used in the WAC-IPWRA estimators.
5. We could modify approach 1 to model the outcome and the time to censoring so that censoring is handled by weighting and its own model instead of by likelihood adjustment. This approach is used in the WRA estimators.

While researcher preferences over what to model largely dictate the approach selected, we quickly note two points that could affect which approach works best. First, we can adjust for censoring by weighting only when censoring time is random. Second, weighting estimators become unstable if the weights get too large.

In the next section, we elaborate on the assumptions needed and the trade-offs among the approaches to estimation.

## Assumptions and trade-offs

The estimators implemented in `stteffects` require three assumptions: conditional independence, sufficient overlap, and correct adjustment for censoring.

### The conditional independence assumption

All estimators implemented in `stteffects` require the potential outcomes to be independent of the treatment assignment after conditioning on the covariates. Randomized experiments and the Heckman selection model are two motivating frameworks for the conditional independence assumption.

When the treatment is assigned randomly, the randomization ensures that the potential outcomes are independent of the treatment assignment. In observational data, the treatment is not randomly assigned. However, many important questions can only be answered using observational data because it would be unethical to randomly allocate hazardous treatments, for example, smoking. The conditional independence assumption in observational data says that treatment assignment is as good as random after conditioning on the covariates.

We can also understand conditional independence from a modeling framework. The Heckman selection model specifies that each of the potential outcomes and the treatment assignment process are functions of observable covariates and unobservable errors. The potential outcomes are conditionally independent of the treatment assignment when the unobservable errors in the treatment-assignment process are independent of the unobservable errors in each of the potential-outcome processes. See *The CI assumption* in [TE] **teffects intro advanced** for a detailed example.

Both frameworks lead to the same conclusion: we need to observe and to condition on a sufficient number of covariates.

Essentially, all the estimators in `stteffects` are equally susceptible to violations of the conditional independence assumption. No one estimator is any more robust to the conditional independence assumption than any other one.

Estimating the ATE among the subpopulation of those who get the treatment requires a significantly weaker version of the CI assumption; see *Assumptions for the ATET* below.

For more details about the conditional independence assumption, see *The CI assumption* in [TE] **teffects intro advanced**, and see Rosenbaum and Rubin (1983), Heckman (1997), Imbens and Wooldridge (2009), Cameron and Trivedi (2005, sec. 25.2), Wooldridge (2010, chap. 21), and Vittinghoff et al. (2012, chap. 9).

## The sufficient overlap assumption

The sufficient overlap assumption requires that each individual have a sufficiently positive probability of being assigned to each treatment level. We believe that the RA estimator is more robust than the other estimators to near violations of the sufficient overlap condition, under correct model specification.

The overlap condition has no specification test, but using `teoverlap` and then summarizing the predicted treatment probabilities presents good diagnostics of overlap problems.

## The correct adjustment for censoring assumption

The correct adjustment for censoring assumption has two parts. First, either the censoring time must be fixed or the process must be conditionally-on-covariates independent of the potential outcomes and the treatment-assignment process. This assumption is standard in survival analysis; see, for example, Kalbfleisch and Prentice (2002, chap. 3).

Second, the method used to adjust to censoring must be correct. For the RA and LAC-IPWRA estimators, which use likelihood-adjusted censoring, the second assumption is no more restrictive than assuming correct specification of the outcome model. For the IPW, WAC-IPWRA, and WRA estimators, which adjust by weighting, the second assumption requires that the censoring be random and that the censoring process be correctly modeled.

Under correct specification, all the estimators in `stteffects` perform well. However, we believe that estimators that use likelihood adjustment instead of weighting are more robust for three reasons.

1. The estimators that use weighting to adjust for censoring cannot handle fixed censoring processes. If the censoring process is not random, the weights are not well defined.
2. The estimators that use weighting to adjust for censoring do not allow the random censoring process to vary by treatment level.
3. The estimators that use weighting to adjust for censoring require an additional sufficient overlap condition: the probability of not being censored must be sufficiently greater than 0 or else the weights that adjust for censoring get too large.

While the estimators that use WAC instead of LAC require a few more assumptions, some researchers are more comfortable modeling the treatment and censoring than the outcome. In this case, the IPW or WAC-IPWRA estimator would be the estimator of choice.

See *Specification diagnostics and tests* below for information about testing these assumptions.

## Assumptions for the ATET

We noted in *Average treatment effect on the treated* that the ATET is sometimes more interesting than the ATE. We can also estimate the ATET under less restrictive versions of the conditional independence assumption and the sufficient overlap assumption than those required for the ATE.

While ATE estimation requires that the potential outcomes for both the treated and the not treated be conditionally independent of treatment assignment, ATET estimation requires that only the not treated potential outcome be conditionally independent of treatment assignment.

This weaker version of conditional independence allows the gains from the treatment to be related to treatment assignment, after conditioning on the covariates. We can estimate the ATET, but not the ATE, if some unobserved factor increases (or decreases) the likelihood of assignment to the treatment, increases (or decreases) the time to event in the treatment group, and has no effect on the time to event when not in the treatment group.

For example, suppose that smoking is an acquired taste and that individuals who acquire the taste for smoking more easily are less adversely affected by smoking and otherwise similar to everyone else when not smoking. Taste for smoking is unobservable, and our data have no measure of this variable. In this case, we could estimate the ATET but not the ATE.

The weaker version of the sufficient overlap assumption only requires that each individual in the treated subpopulation have a positive probability of not getting treated. In contrast, ATE estimation requires that each individual in the population have a positive probability of getting each treatment level. In particular, we can estimate the ATET, but not the ATE, when some individuals in the population have zero chance of getting the treatment. For example, we could estimate the ATET, but not the ATE, if some women will never smoke for religious reasons.

Even when the conditions for ATE estimation hold, the ATE and ATET may differ. Finding that the ATET is significantly different from the ATE does not mean that the ATE is incorrectly estimated.

See Heckman (1997) and Wooldridge (2010, 911–912) for more information about the assumptions necessary to estimate the ATET.

## Specification diagnostics and tests

After `stteffects ipw` and `stteffects ipwra`, some specification checks for the treatment-assignment model and the overlap condition are available.

The checks for the treatment-assignment model are known as balance checks. When the covariate distributions are invariant to the treatment level, the covariates are said to be balanced. The concept of balanced covariates comes from the experimental literature, in which random treatment assignment ensures that the covariates are balanced.

In observational data, the covariates are almost never balanced in the raw data. Weighting methods can be viewed as using a treatment-assignment model to balance the covariates. If the treatment-assignment model is well specified, the weights constructed from this model will balance the covariates. One of the nice features of balance checks is that they do not depend on the outcome or its distribution. This fact is especially useful for survival-time outcomes because censoring of the outcome has no effect on the balance checks, so the balance checks implemented in `tebalance` work without modification.

Conditional on the treatment-assignment model being well specified, we can use the estimated probabilities of treatment, known as the propensity scores, to look for signs that the overlap condition is violated. These checks depend only on the estimated treatment probabilities and are not affected by any censoring of the outcome, so the methods implemented in `teoverlap` work without modification.

We begin examining our model by using `tebalance summarize` after refitting the models used by the LAC-IPWRA estimator.

```
. quietly stteffects ipwra (age exercise diet education)
> (smoke age exercise diet education)

. tebalance summarize
Covariate balance summary
```

	Raw	Weighted
Number of obs =	2,000	2,000.0
Treated obs =	738	994.1
Control obs =	1,262	1,005.9

	Standardized differences		Variance ratio	
	Raw	Weighted	Raw	Weighted
age	-.3122094	-.0184574	.8547308	.9370065
exercise	-.4975269	-.0458412	.4966778	.8342339
diet	-.2479756	.0021802	.7937645	1.095347
education	-.4801442	-.0216366	.6015139	.978078

The weighted standardized differences are much closer to 0 than the raw standardized differences, and the weighted variance ratios are much closer to 1 than the raw variance ratios. These results indicate that the model-based treatment weights balanced the covariates; see [TE] [tebalance](#) and [TE] [tebalance summarize](#) for details.

The diagnostics presented by `tebalance summarize` are not a formal test. However, we can use `tebalance overid` to conduct a formal test of the hypothesis that the weights constructed from the treatment-assignment model balanced the covariates.

```
. tebalance overid
Iteration 0: criterion = .22681884
Iteration 1: criterion = .22692316 (backed up)
Iteration 2: criterion = .23090158
Iteration 3: criterion = .2311461
Iteration 4: criterion = .23256285
Iteration 5: criterion = .23286304
Iteration 6: criterion = .23335858
Iteration 7: criterion = .2335567
Iteration 8: criterion = .2335671
Iteration 9: criterion = .23356711

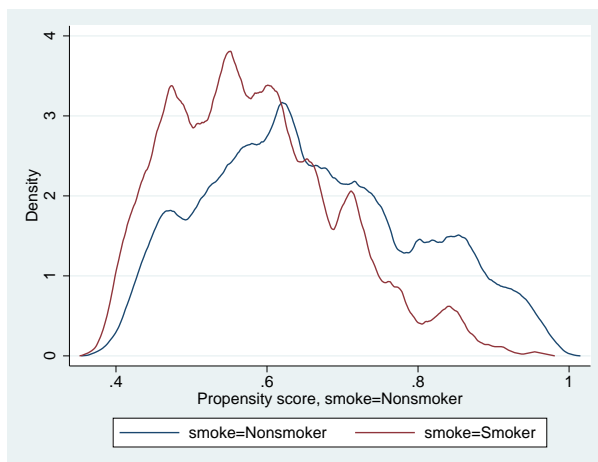
Overidentification test for covariate balance
H0: Covariates are balanced

      chi2(5)      = 3.28142
Prob > chi2      = 0.6567
```

There is no significant evidence against the null hypothesis. The interpretation is that we do not reject the null hypothesis that the treatment-assignment model is well specified; see [TE] [tebalance](#) and [TE] [tebalance overid](#) for details.

Given that we do not reject the treatment-assignment model, we can use this model to look for evidence that the overlap condition is violated. We begin by using `teoverlap`.

```
. teffects overlap, ptlevel(Smoker)
```



The densities of the propensity scores for the smokers and nonsmokers appear to have the same support, indicating that there is no violation of the overlap condition. The only indicator of a possible problem is that the support of the density for nonsmokers gets very close to 0. This problem would affect ATE estimation but not ATET estimation, as discussed in [Assumptions and trade-offs](#). To further investigate, we compute and summarize the predicted propensity score by treatment level.

```
. predict ps1, ps tlevel(Smoker)
. summarize ps1 if smoke == 0
```

Variable	Obs	Mean	Std. dev.	Min	Max
ps1	1,262	.3410001	.1381673	.014819	.6161401

```
. summarize ps1 if smoke == 1
```

Variable	Obs	Mean	Std. dev.	Min	Max
ps1	738	.4168805	.1107557	.0454891	.6216282

To interpret these results, recall that ATE estimation requires that the minimum propensity score for each treatment level be sufficiently greater than 0 and that the maximum propensity score for each treatment level be sufficiently less than 1. Also recall that ATET estimation only requires that the maximum propensity score for each treatment level be sufficiently less than 1.

For ATE estimation, only the minimum predicted propensity score for nonsmokers presents a challenge, and 0.015 is probably not too small. For ATET estimation, neither maximum causes concern.

For information about choosing among the `stteffects` estimators and their functional forms for the different models, see [Model choice](#) under *Remarks and examples* in [\[TE\] teffects intro advanced](#).

## Multivalued treatments

`stteffects` can estimate treatment effects for multivalued treatments; here we provide some examples. See [\[TE\] teffects multivalued](#) for an introduction to interpreting effects from multivalued treatments.



▷ Example 6: Multivalued ATE estimation

We have another fictional dataset that records the time to a second heart attack among women aged 45–55 years. In this dataset, `atime` is the observed time in years to the second heart attack, and `fail` is the 0/1 indicator that a second heart attack was observed and recorded in `atime`. (When `fail` is 1, `atime` records the time to the second attack; when `fail` is 0, `atime` records a censored observation of the time to the second attack.)

These data also contain the age at the time of the first heart attack (`age`), and indices of each woman’s exercise level (`exercise`), diet quality (`diet`), and education attainment (`education`) prior to her first heart attack.

The treatment, smoking, is stored in the categorical variable `smoke`, which has the following value labels. The women who never smoked are labeled as N; the women who previously smoked but quit before their first heart attack are labeled as B; the women who previously smoked but quit after their first heart attack are labeled as A; and the women who continued to smoke after their first heart attack are labeled as S.

We begin by first reading in the data and then reviewing information previously stored using `stset`.

```
. use https://www.stata-press.com/data/r17/sheartm, clear
(Time to second heart attack (fictional))
. stset
-> stset atime, failure(fail)
Survival-time data settings
      Failure event: fail!=0 & fail<.
Observed time interval: (0, atime]
      Exit on or before: failure
```

---

```
10,000 total observations
      0 exclusions
```

---

```
10,000 observations remaining, representing
      9,741 failures in single-record/single-failure data
27,999.155 total analysis time at risk and under observation
                        At risk from t =          0
Earliest observed entry t =          0
Last observed exit t = 17.40826
```

We continue by tabulating the treatment variable `smoke`.

```
. tabulate smoke
```

Smoking level	Freq.	Percent	Cum.
N	3,167	31.67	31.67
B	2,263	22.63	54.30
A	1,924	19.24	73.54
S	2,646	26.46	100.00
Total	10,000	100.00	

We see that 31.67% of the women never smoked, 22.63% of the women previously smoked but quit before their first heart attack, 19.24% of the women previously smoked but quit after their first heart attack, and 26.46% of the women continued to smoke after their first heart attack.

We now use `stteffects ra` to estimate the ATE by RA. We model the outcome as a function of age, exercise, diet, and education, and we specify that `smoke` is the treatment variable.

```
. stteffects ra (age exercise diet education) (smoke)
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 1.390e-20
Iteration 1:  EE criterion = 2.220e-29
Survival treatment-effects estimation      Number of obs      =      10,000
Estimator      : regression adjustment
Outcome model  : Weibull
Treatment model: none
Censoring model: none
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
<b>ATE</b>						
smoke						
(B vs N)	-.4129793	.0317	-13.03	0.000	-.47511	-.3508485
(A vs N)	-1.281031	.032866	-38.98	0.000	-1.345447	-1.216614
(S vs N)	-2.167359	.0338994	-63.93	0.000	-2.233801	-2.100917
<b>POmean</b>						
smoke						
N	3.745919	.0289014	129.61	0.000	3.689273	3.802565

The average time to a second heart attack is 0.41 years sooner when all the women smoked at some point but quit before their first heart attack than when all the women never smoked. The average time to a second heart attack is 1.28 years sooner when all the women smoked at some point but quit after their first heart attack than when all the women never smoked. The average time to a second heart attack is 2.17 years sooner when all the women continued to smoke after their first heart attack than when all the women never smoked.

▷ Example 7: Multivalued ATET estimation

In the at-risk subpopulation of women who continued to smoke, we want to estimate the effect of continuing to smoke (S) versus quitting after the first heart attack (A). Below we estimate the ATETs by RA, specifying A to be the control level and S to be the treatment level.

```
. stteffects ra (age exercise diet education) (smoke), atet control(A) tlevel(S)
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:   EE criterion = 1.390e-20
Iteration 1:   EE criterion = 2.049e-29
Survival treatment-effects estimation      Number of obs      =      10,000
Estimator      : regression adjustment
Outcome model  : Weibull
Treatment model: none
Censoring model: none
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATET						
smoke						
(N vs A)	1.290123	.0377552	34.17	0.000	1.216125	1.364122
(B vs A)	.8748349	.0239595	36.51	0.000	.8278751	.9217946
(S vs A)	-.8869257	.0272301	-32.57	0.000	-.9402958	-.8335557
POmean						
smoke						
A	2.500108	.0217833	114.77	0.000	2.457413	2.542802

The parameter (S vs A) is the one of interest. The estimate implies that the average time to a second heart attack among women who continue to smoke is 0.89 years sooner when they all continue to smoke than when they all quit smoking after their first heart attack.

◀

## Acknowledgments

We thank Xiaofei Bai of Boehringer Ingelheim, Matias Cattaneo of the University of Michigan, Barbara Sianesi of the Institute for Fiscal Studies, and Jeff Wooldridge of Michigan State University for helpful conversations about methods and the literature.

## References

Bai, X., A. A. Tsiatis, and S. M. O’Brien. 2013. Doubly robust estimators of treatment-specific survival distributions in observational studies with stratified sampling. *Biometrics* 69: 830–839. <https://doi.org/10.1111/biom.12076>.

Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.

Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.

Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.

Cleves, M. A., W. W. Gould, and Y. V. Marchenko. 2016. *An Introduction to Survival Analysis Using Stata*. Rev. 3rd ed. College Station, TX: Stata Press.

- Guo, S., and M. W. Fraser. 2015. *Propensity Score Analysis: Statistical Methods and Applications*. 2nd ed. Thousand Oaks, CA: SAGE.
- Heckman, J. 1997. Instrumental variables: A study of implicit behavioral assumptions used in making program evaluations. *Journal of Human Resources* 32: 441–462. <https://doi.org/10.2307/146178>.
- Hirano, K., G. W. Imbens, and G. Ridder. 2003. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica* 71: 1161–1189. <https://doi.org/10.1111/1468-0262.00442>.
- Holland, P. W. 1986. Statistics and causal inference. *Journal of the American Statistical Association* 81: 945–960. <https://doi.org/10.2307/2289064>.
- Imbens, G. W. 2000. The role of the propensity score in estimating dose–response functions. *Biometrika* 87: 706–710. <https://doi.org/10.1093/biomet/87.3.706>.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.
- Kalbfleisch, J. D., and R. L. Prentice. 2002. *The Statistical Analysis of Failure Time Data*. 2nd ed. New York: Wiley.
- Robins, J. M., and A. Rotnitzky. 2006. Inverse probability weighting in survival analysis. In *Survival and Event History Analysis*, ed. N. Keiding and P. K. Andersen, 266–271. Chichester, UK: Wiley.
- Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55. <https://doi.org/10.2307/2335942>.
- Royston, P. 2015. Estimating the treatment effect in a clinical trial using difference in restricted mean survival time. *Stata Journal* 15: 1098–1117.
- Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701. <https://doi.org/10.1037/h0037350>.
- Vittinghoff, E., D. V. Glidden, S. C. Shiboski, and C. E. McCulloch. 2012. *Regression Methods in Biostatistics: Linear, Logistic, Survival, and Repeated Measures Models*. 2nd ed. New York: Springer.
- Wooldridge, J. M. 2007. Inverse probability weighted estimation for general missing data problems. *Journal of Econometrics* 141: 1281–1301. <https://doi.org/10.1016/j.jeconom.2007.02.002>.
- . 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

## Also see

[TE] **stteffects postestimation** — Postestimation tools for stteffects

[TE] **teffects intro advanced** — Advanced introduction to treatment effects for observational data

[ST] **streg** — Parametric survival models

[ST] **stset** — Declare data to be survival-time data

[U] **20 Estimation and postestimation commands**

# Title

**stteffects ipw** — Survival-time inverse-probability weighting

<a href="#">Description</a>	<a href="#">Quick start</a>	<a href="#">Menu</a>	<a href="#">Syntax</a>
<a href="#">Options</a>	<a href="#">Remarks and examples</a>	<a href="#">Stored results</a>	<a href="#">Methods and formulas</a>
<a href="#">References</a>	<a href="#">Also see</a>		

## Description

`stteffects ipw` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational survival-time data with random time to censoring. Estimation is by inverse-probability weighting (IPW). IPW estimators use weighted averages of the observed outcome. The estimated weights correct for missing data on the potential outcomes and for censored survival times. `stteffects ipw` offers several choices for the functional forms of the treatment model and the time-to-censoring model. Binary and multivalued treatments are accommodated.

See [\[TE\] stteffects intro](#) for an overview of estimating treatment effects from observational survival-time data.

## Quick start

Specify `time` as observed failure time and `fail` as failure indicator

```
stset time, failure(fail)
```

ATE of binary `treat2` on `time` by IPW using a logistic model of `treat2` on `x` and `w` and using `x` and `w` in a Weibull model for the censoring time

```
stteffects ipw (treat2 x w) (x w)
```

As above, but estimate the ATET

```
stteffects ipw (treat2 x w) (x w), atet
```

ATE of `treat2` on `time` by IPW using a probit model of `treat2` on `x` and `w` and using `x` and `w` in a gamma model for the censoring time

```
stteffects ipw (treat2 x w, probit) (x w, gamma)
```

ATE for treatment levels 2 and 3 of three-valued treatment `treat3`

```
stteffects ipw (treat3 x w) (x w)
```

As above, and specify that `treat3 = 3` is the control level using the value label “MyControl” for 3

```
stteffects ipw (treat3 x w) (x w), control("MyControl")
```

## Menu

Statistics > Treatment effects > Survival outcomes > Inverse-probability weighting (IPW)

## Syntax

```
stteffects ipw (tvar tmvarlist [, tmoptions]) (cmvarlist [, cmoptions])
  [if] [in] [, stat options]
```

*tvar* must contain integer values representing the treatment levels.

*tmvarlist* specifies the variables that predict treatment assignment in the treatment model.

*cmvarlist* specifies the variables that predict censoring in the censoring model.

<i>tmoptions</i>	Description
------------------	-------------

Model	
<code>logit</code>	logistic treatment model; the default
<code>probit</code>	probit treatment model
<code>hetprobit(<i>varlist</i>)</code>	heteroskedastic probit treatment model
<code>noconstant</code>	suppress constant from treatment model

<i>cmoptions</i>	Description
------------------	-------------

Model	
<code>weibull</code>	Weibull; the default
<code>exponential</code>	exponential
<code>gamma</code>	two-parameter gamma
<code>lnormal</code>	lognormal
<code>ancillary(<i>avarlist</i> [, <code>noconstant</code>])</code>	specify variables used to model ancillary parameter
<code>noconstant</code>	suppress constant from censoring model

<i>stat</i>	Description
-------------	-------------

Stat	
<code>ate</code>	estimate average treatment effect in population; the default
<code>atet</code>	estimate average treatment effect on the treated
<code>pomeans</code>	estimate potential-outcome means

<i>options</i>	Description
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>robust</code> , <code>cluster <i>clustvar</i></code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>aequations</code>	display auxiliary-equation results
<code>noshow</code>	do not show st setting information
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
<code>iterinit(#)</code>	specify starting-value iterations; seldom used
Advanced	
<code>pstolerance(#)</code>	set tolerance for the overlap assumption
<code>osample(<i>newvar</i>)</code>	identify observations that violate the overlap assumption
<code>control(#   <i>label</i>)</code>	specify the level of <i>tvar</i> that is the control
<code>tlevel(#   <i>label</i>)</code>	specify the level of <i>tvar</i> that is the treatment
<code>coeflegend</code>	display legend instead of statistics

You must `stset` your data before using `stteffects`; see [ST] `stset`.

`tmvarlist`, `cmvarlist`, and `avarlist` may contain factor variables; see [U] 11.4.3 Factor variables.

`bootstrap`, `by`, `collect`, `jackknife`, and `statsby` are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the `bootstrap` prefix; see [R] `bootstrap`.

`fweights`, `iweights`, and `pweights` may be specified using `stset`; see *Weights* under *Remarks and examples* in [ST] `stset`. However, weights may not be specified if you are using the `bootstrap` prefix.

`coeflegend` does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

## Options

### Model

`ancillary(avarlist [, noconstant ])` specifies the variables used to model the ancillary parameter.

By default, the ancillary parameter does not depend on covariates. Specifying `ancillary(avarlist, noconstant)` causes the constant to be suppressed in the model for the ancillary parameter.

`noconstant`; see [R] Estimation options.

### Stat

`stat` is one of three statistics: `ate`, `atet`, or `pomeans`. `ate` is the default.

`ate` specifies that the average treatment effect be estimated.

`atet` specifies that the average treatment effect on the treated be estimated.

`pomeans` specifies that the potential-outcome means for each treatment level be estimated.

## SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster clustvar`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce\\_option](#).

## Reporting

`level(#)`; see [R] [Estimation options](#).

`aequations` specifies that the results for the outcome-model or the treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

`noshow` prevents `stteffects ipw` from showing the key `st` variables. This option is rarely used because most people type `stset`, `show` or `stset`, `noshow` to permanently set whether they want to see these variables mentioned at the top of the output of every `st` command; see [ST] [stset](#).

`display_options`: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

## Maximization

`maximize_options`: `iterate(#)`, `[no]log`, and `from(init_specs)`; see [R] [Maximize](#). These options are seldom used.

`init_specs` is one of

`matname [ , skip copy ]`

`# [ , # ... ] , copy`

`iterinit(#)` specifies the maximum number of iterations used to calculate the starting values. This option is seldom used.

## Advanced

`pstolerance(#)` specifies the tolerance used to check the overlap assumption. The default value is `pstolerance(1e-5)`. `stteffects` will exit with an error if an observation has an estimated propensity score smaller than that specified by `pstolerance()`.

`osample(newvar)` specifies that indicator variable `newvar` be created to identify observations that violate the overlap assumption.

`control(#|label)` specifies the level of `tvar` that is the control. The default is the first treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with the statistic `pomeans`. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(#|label)` specifies the level of `tvar` that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

The following option is available with `stteffects` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).



## Remarks and examples

If you are not familiar with the framework for treatment-effects estimation from observational survival-time data, please see [\[TE\] stteffects intro](#).

IPW estimators use contrasts of weighted averages of observed outcomes to estimate treatment effects. The estimated weights correct for data that are missing because each subject is only observed after receiving one of the possible treatment levels and because some survival-time outcomes are censored.

The IPW estimators implemented in `stteffects ipw` use a three-step approach to estimating the ATE:

1. Estimate the parameters of a treatment-assignment model, and compute the component of the estimated weights that accounts for data missing because each subject is only observed after receiving one of the possible treatment levels.
2. Estimate the parameters of a time-to-censoring model, and compute the component of the estimated weights that accounts for data lost to censoring.
3. Use the estimated weights to compute weighted averages of the outcomes for each treatment level.

To estimate the ATET, we use different weights in step 2.

The time to censoring must be random to use `stteffects ipw` because the model in step 2 is not well defined if the time to censoring is fixed. See [\[TE\] stteffects intro](#) for more details. For information about estimators that accommodate a fixed time to censoring, see [\[TE\] stteffects ra](#) and [\[TE\] stteffects ipwra](#).

Here we note only a few entry points to the vast literature on IPW estimators. [Hirano, Imbens, and Ridder \(2003\)](#), [Imbens \(2000, 2004\)](#), [Imbens and Wooldridge \(2009\)](#), [Rosenbaum and Rubin \(1983\)](#), [Robins and Rotnitzky \(2006\)](#), [Wooldridge \(2002, 2007\)](#), [Cameron and Trivedi \(2005, chap. 25\)](#), [Wooldridge \(2010, chap. 21\)](#), and [Vittinghoff et al. \(2012, chap. 9\)](#) provide excellent general introductions to estimating ATEs and to the IPW estimators in particular.

Like `streg` and other survival-time commands, `stteffects ipw` uses the outcome variable and the failure indicator computed by, and optionally weights specified with, `stset`. `stteffects ipw` is not appropriate for data with time-varying covariates, also known as multiple-record survival-time data, or for delayed-entry data.

### ► Example 1: Estimating the ATE

Suppose we wish to study the effect of smoking on the time to a second heart attack among women aged 45–55 years. In our fictional `sheart` dataset, `atime` is the observed time in years to a second heart attack or censoring, and `fail` is the 0/1 indicator that a second heart attack was observed. (When `fail` is 1, `atime` records the time to the second heart attack; when `fail` is 0, `atime` records a censored observation of the time to a second heart attack.) We previously `stset` these data; see [A quick tour of the estimators](#) in [\[TE\] stteffects intro](#).

The treatment, smoking, is stored in the 0/1 indicator `smoke`. These data also contain age at the time of the first heart attack (`age`), and indices of the level of exercise (`exercise`), diet quality (`diet`), and education (`education`) prior to the first heart attack.

We can use `stteffects ipw` to estimate the ATE. We model treatment assignment using the default logit model with covariates on `age`, `exercise`, and `education`. We model the time to censoring using the default Weibull model with covariates on `age`, `exercise`, `diet`, and `education`.

```

. use https://www.stata-press.com/data/r17/sheart
(Time to second heart attack (fictional))
. stteffects ipw (smoke age exercise education) (age exercise diet education)
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 2.042e-18
Iteration 1:  EE criterion = 5.191e-31
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: logit
Censoring model: Weibull

```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE smoke (Smoker vs Nonsmoker)	-2.22226	.6307573	-3.52	0.000	-3.458522	-.9859983
POMean smoke Nonsmoker	4.235569	.5210937	8.13	0.000	3.214244	5.256894

When every woman smoked in the population of women aged 45–55 years who have had a heart attack, the average time to a second heart attack is estimated to be 2.22 years less than when no women in the population of interest smoked. The estimated average time to a second heart attack when no women in the population of interest smoked is 4.24 years.

The ratio of the ATE to the control-level POM measures the importance of the effect. In this example, when every woman smoked, the average time to a second heart attack falls by an estimated 52% relative to the case when none of them smoked. See [example 3](#) in [TE] [stteffects ra](#) for an example that uses `nlcom` to compute a point estimate and a confidence interval for this ratio.

◀

## ► Example 2: Different treatment and censoring models

Instead of a logit model for the treatment assignment, we could have used a probit or a heteroskedastic probit model. Instead of a Weibull model for the censoring time, we could have used an exponential, a gamma, or a lognormal model. For a quick comparison, we now estimate the ATE using a probit model for the treatment assignment and using a gamma model for the censoring time.

```
. stteffects ipw (smoke age exercise education, probit)
> (age exercise diet education, gamma)

      Failure _d: fail
      Analysis time _t: atime
Iteration 0:   EE criterion = 3.534e-15
Iteration 1:   EE criterion = 5.263e-27
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: probit
Censoring model: gamma
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE smoke (Smoker vs Nonsmoker)	-2.646808	.8368254	-3.16	0.002	-4.286956	-1.006661
POmean smoke Nonsmoker	4.702301	.7404567	6.35	0.000	3.251033	6.15357

The estimated ATE of  $-2.65$  and control-level POM of  $4.70$  are similar to the values of  $-2.22$  and  $4.24$  reported in [example 1](#).



### ► Example 3: Estimating the ATET

Intuitively, the ATET measures the effect of the treatment on an at-risk subpopulation. Sometimes the subpopulation that gets the treatment defines such an at-risk subpopulation. The ATET has the added benefit that it can be estimated under weaker conditions than the ATE; see [Assumptions and trade-offs](#) under *Remarks and examples* in [\[TE\] stteffects intro](#).

```
. stteffects ipw (smoke age exercise education) (age exercise diet education),
> atet
```

```
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:   EE criterion = 2.042e-18
Iteration 1:   EE criterion = 1.248e-32
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: logit
Censoring model: Weibull
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATET						
smoke (Smoker vs Nonsmoker)	-1.846136	.5076872	-3.64	0.000	-2.841185	-.8510877
POmean						
smoke Nonsmoker	3.543788	.474395	7.47	0.000	2.613991	4.473585

When every woman in the subpopulation smoked, the average time to a second heart attack is estimated to be 1.85 years less than when no women in the subpopulation smoked. The estimated average time to a second heart attack when no women in the subpopulation smoked is 3.54 years.

◀

## Stored results

`stteffects ipw` stores the following in `e()`:

### Scalars

```
e(N)           number of observations
e(nj)          number of observations for treatment level j
e(N_clust)     number of clusters
e(k_eq)        number of equations in e(b)
e(k_levels)    number of levels in treatment variable
e(treated)     level of treatment variable defined as treated
e(control)     level of treatment variable defined as control
e(converged)   1 if converged, 0 otherwise
```

### Macros

```
e(cmd)         stteffects
e(cmdline)     command as typed
e(dead)        _d
e(depvar)      _t
e(tvar)        name of treatment variable
e(subcmd)      ipw
e(tmodel)      treatment model: logit, probit, or hetprobit
e(cmodel)      censoring model: weibull, exponential, gamma, or lognormal
e(stat)        statistic estimated: ate, atet, or pomeans
e(wtype)       weight type
e(wexp)        weight expression
e(title)       title in estimation output
e(clustvar)    name of cluster variable
e(tlevels)     levels of treatment variable
```

<code>e(vce)</code>	<code>vcetype</code> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>
Matrices	
<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators
Functions	
<code>e(sample)</code>	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any *r*-class command is run after the estimation command.

## Methods and formulas

The methods and formulas for the IPW estimators implemented in `stteffects ipw` are given in *Methods and formulas* of [TE] `stteffects ipwra`.

## References

- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.
- Bai, X., A. A. Tsiatis, and S. M. O'Brien. 2013. Doubly robust estimators of treatment-specific survival distributions in observational studies with stratified sampling. *Biometrics* 69: 830–839. <https://doi.org/10.1111/biom.12076>.
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.
- Guo, S., and M. W. Fraser. 2015. *Propensity Score Analysis: Statistical Methods and Applications*. 2nd ed. Thousand Oaks, CA: SAGE.
- Hirano, K., G. W. Imbens, and G. Ridder. 2003. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica* 71: 1161–1189. <https://doi.org/10.1111/1468-0262.00442>.
- Imbens, G. W. 2000. The role of the propensity score in estimating dose–response functions. *Biometrika* 87: 706–710. <https://doi.org/10.1093/biomet/87.3.706>.
- . 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics* 86: 4–29. <https://doi.org/10.1162/003465304323023651>.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.
- Robins, J. M., and A. Rotnitzky. 2006. Inverse probability weighting in survival analysis. In *Survival and Event History Analysis*, ed. N. Keiding and P. K. Andersen, 266–271. Chichester, UK: Wiley.

- Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55. <https://doi.org/10.2307/2335942>.
- Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701. <https://doi.org/10.1037/h0037350>.
- Tsiatis, A. A. 2006. *Semiparametric Theory and Missing Data*. New York: Springer.
- Vittinghoff, E., D. V. Glidden, S. C. Shiboski, and C. E. McCulloch. 2012. *Regression Methods in Biostatistics: Linear, Logistic, Survival, and Repeated Measures Models*. 2nd ed. New York: Springer.
- Wooldridge, J. M. 2002. Inverse probability weighted M-estimators for sample selection, attrition, and stratification. *Portuguese Economic Journal* 1: 117–139. <https://doi.org/10.1007/s10258-002-0008-x>.
- . 2007. Inverse probability weighted estimation for general missing data problems. *Journal of Econometrics* 141: 1281–1301. <https://doi.org/10.1016/j.jeconom.2007.02.002>.
- . 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

## Also see

- [TE] **stteffects postestimation** — Postestimation tools for stteffects
- [TE] **stteffects intro** — Introduction to treatment effects for observational survival-time data
- [ST] **streg** — Parametric survival models
- [ST] **stset** — Declare data to be survival-time data
- [U] **20 Estimation and postestimation commands**

<a href="#">Description</a>	<a href="#">Quick start</a>	<a href="#">Menu</a>	<a href="#">Syntax</a>
<a href="#">Options</a>	<a href="#">Remarks and examples</a>	<a href="#">Stored results</a>	<a href="#">Methods and formulas</a>
<a href="#">References</a>	<a href="#">Also see</a>		

## Description

`stteffects ipwra` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational survival-time data by inverse-probability-weighted regression adjustment (IPWRA). IPWRA estimators use missingness-adjusted regression coefficients to compute averages of treatment-level predicted outcomes. Contrasts of these averages estimate the treatment effects. `stteffects ipwra` offers several choices for the functional forms of the outcome model, of the treatment model, and of the optional time-to-censoring model. Binary and multivalued treatments are accommodated.

See [\[TE\] stteffects intro](#) for an overview of estimating treatment effects from observational survival-time data.

## Quick start

Specify time as observed failure time and fail as failure indicator

```
stset time, failure(fail)
```

ATE of binary treatment `treat2` estimated by IPWRA using a Weibull model for time on `x1` and `x2` and a logistic model for `treat2` on `x1` and `w`

```
stteffects ipwra (x1 x2) (treat2 x1 w)
```

As above, but estimate the ATET

```
stteffects ipwra (x1 x2) (treat2 x1 w), atet
```

Gamma model for time and probit model for `treat2`

```
stteffects ipwra (x1 x2, gamma) (treat2 x1 w, probit)
```

ATE for each level of three-valued treatment `treat3`

```
stteffects ipwra (x1 x2) (treat3 x1 w)
```

As above, and specify that `treat3 = 3` is the control level using the value label "MyControl" for 3

```
stteffects ipwra (x1 x2) (treat3 x1 w), control("MyControl")
```

ATE of `treat2` estimated by IPWRA using a Weibull model for time on `x1` and `x2`, a logistic model for `treat2` on `x1` and `w`, and a Weibull model for the time to censoring with covariates `x1` and `x2`

```
stteffects ipwra (x1 x2) (treat2 x1 w) (x1 x2)
```

Gamma model for time, probit model for `treat2`, and gamma model for censoring

```
stteffects ipwra (x1 x2, gamma) (treat2 x1 w, probit) (x1 x2, gamma)
```

## Menu

Statistics > Treatment effects > Survival outcomes > Regression adjustment with IPW

## Syntax

```
stteffects ipwra (omvarlist [, omoptions]) (tvar tmvarlist [, tmoptions])
  [(cmvarlist [, cmoptions])] [if] [in] [, stat options]
```

*omvarlist* specifies the variables that predict the survival-time variable in the outcome model.

*tvar* must contain integer values representing the treatment levels.

*tmvarlist* specifies the variables that predict treatment assignment in the treatment model.

*cmvarlist* specifies the variables that predict censoring in the censoring model.

<i>omoptions</i>	Description
Model	
<a href="#">weibull</a>	Weibull; the default
<a href="#">exponential</a>	exponential
<a href="#">gamma</a>	two-parameter gamma
<a href="#">lnormal</a>	lognormal
<a href="#">ancillary</a> ( <i>avarlist</i> [, <a href="#">noconstant</a> ])	specify variables used to model ancillary parameter
<a href="#">noconstant</a>	suppress constant from outcome model

<i>tmoptions</i>	Description
Model	
<a href="#">logit</a>	logistic treatment model; the default
<a href="#">probit</a>	probit treatment model
<a href="#">hetprobit</a> ( <i>varlist</i> )	heteroskedastic probit treatment model
<a href="#">noconstant</a>	suppress constant from treatment model

<i>cmoptions</i>	Description
Model	
<a href="#">weibull</a>	Weibull; the default
<a href="#">exponential</a>	exponential
<a href="#">gamma</a>	two-parameter gamma
<a href="#">lnormal</a>	lognormal
<a href="#">ancillary</a> ( <i>avarlist</i> [, <a href="#">noconstant</a> ])	specify variables used to model ancillary parameter
<a href="#">noconstant</a>	suppress constant from censoring model



<i>stat</i>	Description
Stat	
<code>ate</code>	estimate average treatment effect in population; the default
<code>atet</code>	estimate average treatment effect on the treated
<code>pomeans</code>	estimate potential-outcome means
options	
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>robust</code> , <code>cluster <i>clustvar</i></code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>aequations</code>	display auxiliary-equation results
<code>noshow</code>	do not show st setting information
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
<code>iterinit(#)</code>	specify starting-value iterations; seldom used
Advanced	
<code>pstolerance(#)</code>	set tolerance for the overlap assumption
<code>osample(<i>newvar</i>)</code>	identify observations that violate the overlap assumption
<code>control(#   <i>label</i>)</code>	specify the level of <i>tvar</i> that is the control
<code>tlevel(#   <i>label</i>)</code>	specify the level of <i>tvar</i> that is the treatment
<code>coeflegend</code>	display legend instead of statistics

You must `stset` your data before using `stteffects`; see [ST] `stset`.

`omvarlist`, `tmvarlist`, `cmvarlist`, and `avarlist` may contain factor variables; see [U] 11.4.3 Factor variables.

`bootstrap`, `by`, `collect`, `jackknife`, and `statsby` are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the `bootstrap` prefix; see [R] `bootstrap`.

`fweights`, `iweights`, and `pweights` may be specified using `stset`; see *Weights* under *Remarks and examples* in [ST] `stset`. However, weights may not be specified if you are using the `bootstrap` prefix.

`coeflegend` does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

## Options

Model

`ancillary(avarlist [, noconstant ])` specifies the variables used to model the ancillary parameter.

By default, the ancillary parameter does not depend on covariates. Specifying `ancillary(avarlist, noconstant)` causes the constant to be suppressed in the model for the ancillary parameter.

`ancillary()` may be specified for the model for survival-time outcome, for the model for the censoring variable, or for both. If `ancillary()` is specified for both, the varlist used for each model may be different.

`noconstant`; see [R] [Estimation options](#).

## Stat

`stat` is one of three statistics: `ate`, `atet`, or `pomeans`. `ate` is the default.

`ate` specifies that the average treatment effect be estimated.

`atet` specifies that the average treatment effect on the treated be estimated.

`pomeans` specifies that the potential-outcome means for each treatment level be estimated.

## SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster clustvar`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce\\_option](#).

## Reporting

`level(#)`; see [R] [Estimation options](#).

`aequations` specifies that the results for the outcome-model or treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

`noshow` prevents `stteffects ipwra` from showing the key `st` variables. This option is rarely used because most people type `stset`, `show` or `stset`, `noshow` to permanently set whether they want to see these variables mentioned at the top of the output of every `st` command; see [ST] [stset](#).

`display_options`: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

## Maximization

`maximize_options`: `iterate(#)`, `[no]log`, and `from(init_specs)`; see [R] [Maximize](#). These options are seldom used.

`init_specs` is one of

`matname [ , skip copy ]`

`# [ , # ... ] , copy`

`iterinit(#)` specifies the maximum number of iterations used to calculate the starting values. This option is seldom used.

## Advanced

`pstolerance(#)` specifies the tolerance used to check the overlap assumption. The default value is `pstolerance(1e-5)`. `stteffects` will exit with an error if an observation has an estimated propensity score smaller than that specified by `pstolerance()`.

`osample(newvar)` specifies that indicator variable `newvar` be created to identify observations that violate the overlap assumption.

`control(# | label)` specifies the level of *tvar* that is the control. The default is the first treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with the statistic `pomeans`. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(# | label)` specifies the level of *tvar* that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

The following option is available with `stteffects` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

## Remarks and examples

If you are not familiar with the framework for treatment-effects estimation from observational survival-time data, please see [TE] [stteffects intro](#).

IPWRA estimators use estimated weights to obtain missingness-adjusted outcome-regression parameters. The missingness-adjusted outcome-regression parameters are used to compute averages of treatment-level predicted outcomes. Contrasts of these averages estimate the treatment effects.

The estimated weights account for the missing potential outcome and, optionally, for data lost to censoring. The weights are estimated using a treatment-assignment model and, optionally, a model for the censoring time. A term in the estimator for the outcome-regression parameters accounts for data lost to censoring when estimated weights are not used.

There are two versions of the IPWRA estimator because there are two methods of accounting for the data lost to censoring.

1. IPWRA estimators that adjust for censoring by including a term in the likelihood function for the outcome-model parameters are known as likelihood-adjusted-censoring IPWRA (LAC-IPWRA) estimators.
2. IPWRA estimators that adjust for censoring by weighting the likelihood function for the outcome-model parameters by estimated inverse-probability-of-censoring weights are known as weighted-adjusted-censoring IPWRA (WAC-IPWRA) estimators.

The LAC-IPWRA estimators require fewer assumptions than the WAC-IPWRA estimators. Outlining the steps performed by LAC-IPWRA and WAC-IPWRA estimators allows us to be more specific about the trade-offs between the estimators.

LAC-IPWRA estimators use a three-step approach to estimating treatment effects:

1. Estimate the parameters of a treatment-assignment model and compute inverse-probability-of-treatment weights.
2. Obtain the treatment-specific predicted mean outcomes for each subject by using the weighted maximum likelihood estimators. Estimated inverse-probability-of-treatment weights are used to weight the maximum likelihood estimator. A term in the likelihood function adjusts for right-censored survival times.
3. Compute the means of the treatment-specific predicted mean outcomes. Contrasts of these averages provide the estimates of the ATEs. By restricting the computations of the means to the subset of treated subjects, we can obtain the ATETs.

WAC-IPWRA estimators use a four-step approach to estimating treatment effects:

1. Estimate the parameters of a treatment-assignment model and compute inverse-probability-of-treatment weights.
2. Estimate the parameters of a time-to-censoring model and compute inverse-probability-of-censoring weights.
3. Obtain the treatment-specific predicted mean outcomes for each subject by using the weighted maximum likelihood estimators. Estimated inverse-probability-of-treatment weights and inverse-probability-of-censoring weights are used to weight the maximum likelihood estimator. The inverse-probability-of-censoring weights account for right-censored survival times.
4. Compute the means of the treatment-specific predicted mean outcomes. Contrasts of these averages provide the estimates of the ATEs. By restricting the computations of the means to the subset of treated subjects, we can obtain the ATETs.

The WAC-IPWRA estimators require that the censoring time be random and that the time-to-censoring model be well specified. The implemented WAC-IPWRA estimators also require that the time-to-censoring process not vary by treatment level. The LAC-IPWRA estimators do not require these extra assumptions because they use a likelihood term instead of weights to adjust for the data lost to censoring.

Here we note only a few entry points to the vast literature on estimators that combine IPW and RA methods. [Hirano, Imbens, and Ridder \(2003\)](#), [Imbens \(2000, 2004\)](#), [Imbens and Wooldridge \(2009\)](#), [Rosenbaum and Rubin \(1983\)](#), [Robins and Rotnitzky \(1995, 2006\)](#), [Robins, Rotnitzky, and Zhao \(1995\)](#), [Wooldridge \(2002, 2007\)](#), [Cameron and Trivedi \(2005, chap. 25\)](#), [Wooldridge \(2010, chap. 21\)](#), and [Vittinghoff et al. \(2012, chap. 9\)](#) provide excellent general introductions to estimating ATEs and to the IPWRA estimators in particular.

Like `streg` and other survival-time commands, `stteffects ipwra` uses the outcome variable and the failure indicator computed by, and optionally weights specified with, `stset`. `stteffects ipwra` is not appropriate for data with time-varying covariates, also known as multiple-record survival-time data, or for delayed-entry data.

## ► Example 1: Estimating the ATE by LAC-IPWRA

Suppose we wish to study the effect of smoking on the time to a second heart attack among women aged 45–55 years. In our fictional `sheart` dataset, `atime` is the observed time in years to a second heart attack or censoring, and `fail` is the 0/1 indicator that a second heart attack was observed. (When `fail` is 1, `atime` records the time to the second heart attack; when `fail` is 0, `atime` records a censored observation of the time to a second heart attack.) We previously `stset` these data; see [A quick tour of the estimators](#) in [\[TE\] stteffects intro](#).

The treatment, smoking, is stored in the 0/1 indicator `smoke`. These data also contain age at the time of the first heart attack (`age`), and indices of the level of exercise (`exercise`), diet quality (`diet`), and education (`education`) prior to the first heart attack.

We can use `stteffects ipwra` to estimate the ATE. We model the mean survival time using the default Weibull model, controlling for `age`, `exercise`, `diet`, and `education`. We model treatment assignment using the default logit model with covariates `age`, `exercise`, and `education`. We do not specify a time-to-censoring model so that we obtain the LAC estimator.

```
. use https://www.stata-press.com/data/r17/sheart
(Time to second heart attack (fictional))
. stteffects ipwra (age exercise diet education) (smoke age exercise education)
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:   EE criterion = 7.439e-15
Iteration 1:   EE criterion = 1.756e-26
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : IPW regression adjustment
Outcome model  : Weibull
Treatment model: logit
Censoring model: none
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE smoke (Smoker vs Nonsmoker)	-1.591874	.4837332	-3.29	0.001	-2.539973	-.643774
POmean smoke Nonsmoker	4.214263	.2598689	16.22	0.000	3.704929	4.723597

When every woman smoked in the population of women aged 45–55 years who have had a heart attack, the average time to a second heart attack is estimated to be 1.59 years less than when no women in the population of interest smoked. The estimated average time to a second heart attack when no women in the population of interest smoked is 4.21 years.

The ratio of the ATE to the control-level potential-outcome mean (POM) measures the importance of the effect. In this example, when all women smoked, the time to the second heart attack falls by an estimated 38% relative to the case in which no women smoked. See [example 3](#) in [\[TE\] stteffects ra](#) for an example that uses `nlcom` to compute a point estimate and a confidence interval for this ratio.

## ▷ Example 2: Different outcome and treatment models

Instead of a Weibull model for the outcome model, we could have used an exponential, a gamma, or a lognormal model. Instead of a logit model for the treatment assignment, we could have used a probit or a heteroskedastic probit model. This example uses a gamma model for the outcome and a probit model for the treatment assignment.

```
. stteffects ipwra (age exercise diet education, gamma)
> (smoke age exercise education, probit)

      Failure _d: fail
      Analysis time _t: atime

Iteration 0:  EE criterion = 2.644e-13
Iteration 1:  EE criterion = 2.153e-23

Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : IPW regression adjustment
Outcome model  : gamma
Treatment model: probit
Censoring model: none
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE smoke (Smoker vs Nonsmoker)	-1.387303	.4786032	-2.90	0.004	-2.325348	-.4492583
POMean smoke Nonsmoker	3.97986	.2258474	17.62	0.000	3.537207	4.422512

The estimated ATE of  $-1.39$  and control-level POM of  $3.98$  are similar to the values of  $-1.59$  and  $4.21$  that we obtained in [example 1](#).

◀

▷ Example 3: Estimating the ATE by WAC-IPWRA

Rather than using LAC, we may want to specify a time-to-censoring model. We now use `stteffects ipwra` to estimate the ATE by WAC-IPWRA. We use the same specification of the outcome and treatment models that we used in [example 1](#). However, now we specify a time-to-censoring model, using the default Weibull model with covariates `age`, `exercise`, `diet`, and `education`.

```
. stteffects ipwra (age exercise diet education) (smoke age exercise education)
> (age exercise diet education)

      Failure _d: fail
      Analysis time _t: atime

Iteration 0:  EE criterion = 3.673e-18
Iteration 1:  EE criterion = 3.383e-31

Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : IPW regression adjustment
Outcome model  : Weibull
Treatment model: logit
Censoring model: Weibull
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE smoke (Smoker vs Nonsmoker)	-2.285057	.7318456	-3.12	0.002	-3.719448	-.8506656
POMean smoke Nonsmoker	4.385841	.6427521	6.82	0.000	3.12607	5.645612

The estimated ATE of  $-2.29$  differs from the ATE of  $-1.59$  estimated by LAC-IPWRA, but the estimates of the control-level POM are similar between the two models: 4.39 for the WAC compared with 4.21 for the LAC.



## ▷ Example 4: Estimating the ATET by LAC-IPWRA

Intuitively, the ATET measures the effect of the treatment on an at-risk subpopulation. Sometimes the subpopulation that gets the treatment defines such an at-risk subpopulation. The ATET has the added benefit that it can be estimated under weaker conditions than the ATE; see *Assumptions and trade-offs* under *Remarks and examples* in [TE] **stteffects intro**.

```
. stteffects ipwra (age exercise diet education)
> (smoke age exercise education), atet

      Failure _d: fail
      Analysis time _t: atime

Iteration 0:  EE criterion = 2.815e-19
Iteration 1:  EE criterion = 6.331e-31

Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : IPW regression adjustment
Outcome model  : Weibull
Treatment model: logit
Censoring model: none
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATET						
smoke (Smoker vs Nonsmoker)	-1.775107	.3437506	-5.16	0.000	-2.448846	-1.101368
P0mean						
smoke Nonsmoker	4.062424	.2779877	14.61	0.000	3.517578	4.60727

When all women in the subpopulation smoked, the average time to a second heart attack is estimated to be 1.78 years less than when no women in the subpopulation of interest smoked. If no women in the subpopulation of interest smoked, the average time to a second heart attack is 4.06 years.

◀

**Stored results**

**stteffects ipwra** stores the following in `e()`:

## Scalars

```
e(N)           number of observations
e(nj)          number of observations for treatment level j
e(N_clust)     number of clusters
e(k_eq)        number of equations in e(b)
e(k_levels)    number of levels in treatment variable
e(treated)     level of treatment variable defined as treated
e(control)     level of treatment variable defined as control
e(converged)   1 if converged, 0 otherwise
```

## Macros

```
e(cmd)         stteffects
e(cmdline)     command as typed
e(dead)        _d
e(depvar)      _t
e(tvar)        name of treatment variable
```



<code>e(subcmd)</code>	<code>ipwra</code>
<code>e(omodel)</code>	outcome model: weibull, exponential, gamma, or lognormal
<code>e(tmodel)</code>	treatment model: logit, probit, or hetprobit
<code>e(cmodel)</code>	censoring model: weibull, exponential, gamma, or lognormal (if specified)
<code>e(stat)</code>	statistic estimated: ate, atet, or pomeans
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<i>vce</i> type specified in <code>vce()</code>
<code>e(vctype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>
Matrices	
<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators
Functions	
<code>e(sample)</code>	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r`-class command is run after the estimation command.

## Methods and formulas

Methods and formulas are presented under the following headings:

- [Introduction](#)
- [Regression-adjusted estimators](#)
- [Weighted-adjusted-censoring assumptions](#)
- [Weighted regression-adjusted estimators](#)
- [Inverse-probability-weighted estimators](#)
  - [Uncensored data](#)
  - [Inverse-probability-weighted regression-adjustment estimators](#)
    - [Weighted-adjusted-censoring IPWRA](#)
    - [Likelihood-adjusted-censoring IPWRA](#)
  - [Functional-form details](#)

## Introduction

This section presents the methods and formulas used by the estimators implemented in `stteffects ra`, `stteffects wra`, `stteffects ipw`, and `stteffects ipwra`. This section assumes that you are familiar with the concepts and intuition from the estimators discussed in [TE] [teffects intro advanced](#).

Each of the estimators implemented in `stteffects` has a multistep logic but is implemented as one step by simultaneously solving the estimating equations that define each step. This one-step estimating-equation approach provides consistent point estimates and a consistent variance–covariance of the estimator (VCE); see Newey (1984), Wooldridge (2010), and Drukker (2014).

Survival-time treatment-effects estimators handle two types of missing data. First, only one of the potential outcomes is observed, as is standard in causal inference. Second, the potential outcome for the received treatment may be censored. The data missing because of censoring may be handled by an outcome model, a censoring model, or both, just like the data missing due to observing only one potential outcome.

## □ Technical note

Delayed entry would be a third type of missing data. The left-truncation process caused by delayed entry would also need to be modeled to estimate ATE parameters. The estimators implemented in `stteffects` do not allow for delayed entry because they do not have a method for modeling how the left-truncation process selects the sample, conditional on the covariates. □

All the implemented estimators are combinations of regression-adjustment (RA) and inverse-probability-weighted (IPW) techniques. RA estimators use an outcome model to account for the missing potential outcome and for censoring. IPW estimators use models for treatment assignment and censoring to construct weights that account for the missing potential outcome and for censoring.

The remainder of this section provides technical details about how the estimators in `stteffects` were implemented. We provide details only for the two-treatment-level case to simplify the formulas. We provide outlines for how the extensions to the multiple-treatment-level case were implemented.

## Regression-adjusted estimators

We begin with the RA estimators implemented in `stteffects ra`. The RA estimators have the following logic:

- RA1. For each treatment level  $\tau \in \{0, 1\}$ , estimate by maximum likelihood (ML) the parameters  $\beta_\tau$  of a parametric model for the survival-time outcome  $t$  in which  $F(t|\mathbf{x}, \tau, \beta_\tau)$  is the distribution of  $t$  conditional on covariates  $\mathbf{x}$  and treatment level  $\tau$ . Denote the estimates  $\beta_\tau$  by  $\hat{\beta}_{\text{ra}, \tau}$ .
- RA2. Use the estimated  $\hat{\beta}_{\text{ra}, \tau}$  and the functional form implied by  $F(t|\mathbf{x}, \tau, \beta_\tau)$  to estimate the mean survival time, conditional on  $\mathbf{x}$  and treatment level  $\tau$ , for each sample observation, denoted by  $\hat{E}(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{ra}, \tau})$ . Conditional independence of the treatment and the survival-time potential outcomes ensures that  $E(t|\mathbf{x}, \tau, \beta_\tau) = E(t_\tau|\mathbf{x}, \beta_\tau)$ , where  $t_\tau$  is the potential survival-time outcome corresponding to treatment level  $\tau$ . Under correct model specification, sample averages of  $\hat{E}(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{ra}, \tau})$  consistently estimate the POM for treatment level  $\tau$ , denoted by  $\text{POM}_\tau$ .
- RA3. A contrast of the estimated POMs estimates the ATE.

If estimating an ATET, step RA2 is modified to use only the treated observations when estimating the POMs. A contrast of these POMs then estimates the ATET.

The contribution of the  $i$ th observation to the log likelihood that is maximized in step RA1 is

$$L_{\text{ra}}(t_i, \mathbf{x}_i, \tau, \hat{\beta}_{\text{ra}, \tau}) = \varpi_i(\tau_i == \tau) \left[ (1 - c_i) \ln\{f(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{ra}, \tau})\} + c_i \ln\{1 - F(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{ra}, \tau})\} \right] \quad (1)$$

where  $\varpi_i$  is the observation-level weight,  $c_i$  is the 0/1 indicator for whether the survival-time observation on person  $i$  was censored, and  $f(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ra},\tau})$  is the density corresponding to distribution  $F(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ra},\tau})$ . The first term inside the curly braces in (1) accounts for the noncensored observations, and the second term inside the curly braces accounts for the censored observations.

The RA estimators for the POMs simultaneously solve estimating equations (2a) through (2d) for  $\widehat{\beta}_{\text{ra},0}$ ,  $\widehat{\beta}_{\text{ra},1}$ ,  $\widehat{\text{POM}}_{\text{ra},0}$ , and  $\widehat{\text{POM}}_{\text{ra},1}$ .

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ra}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ra},0}, F) = \mathbf{0} \quad (2a)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ra}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ra},1}, F) = \mathbf{0} \quad (2b)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i|\mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ra},0}) - \widehat{\text{POM}}_{\text{ra},0} \right\} = 0 \quad (2c)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i|\mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ra},1}) - \widehat{\text{POM}}_{\text{ra},1} \right\} = 0 \quad (2d)$$

where

$\mathbf{s}_{\text{ra}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ra},0}, F) = \frac{\partial L_{\text{ra}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ra},0})}{\partial \widehat{\beta}_{\text{ra},0}}$  is the vector of score equations from the ML estimator

for  $\widehat{\beta}_{\text{ra},0}$  based on survival-time model  $F$ ,

$\mathbf{s}_{\text{ra}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ra},1}, F) = \frac{\partial L_{\text{ra}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ra},1})}{\partial \widehat{\beta}_{\text{ra},1}}$  is the vector of score equations from the ML estimator

for  $\widehat{\beta}_{\text{ra},1}$  based on survival-time model  $F$ ,

$\widehat{E}(t_i|\mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ra},0})$  is the predicted mean survival time assuming treatment level 0 for observation  $i$  conditional on  $\mathbf{x}_i$ , and

$\widehat{E}(t_i|\mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ra},1})$  is the predicted mean survival time assuming treatment level 1 for observation  $i$  conditional on  $\mathbf{x}_i$ .

The ATE is estimated by replacing (2d) with

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i|\mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ra},1}) - \widehat{\text{POM}}_{\text{ra},0} - \widehat{\text{ATE}}_{\text{ra}} \right\} = 0 \quad (3)$$

and the ATET is estimated by replacing (2c) and (3) with

$$1/N_1 \sum_{i=1}^N \varpi_i (\tau_i == 1) \left\{ \widehat{E}(t_i|\mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ra},0}) - \widehat{\text{POM}}_{\text{ra},\text{cot},0} \right\} = 0$$

$$1/N_1 \sum_{i=1}^N \varpi_i (\tau_i == 1) \left\{ \widehat{E}(t_i|\mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ra},1}) - \widehat{\text{POM}}_{\text{ra},\text{cot},0} - \widehat{\text{ATET}}_{\text{ra}} \right\} = 0$$

where  $N_1 = \sum_{i=1}^N (t_i == 1)$  and  $\widehat{\text{POM}}_{\text{ra},\text{cot},0}$  is the estimated conditional-on-treatment POM for treatment level 0.

Asymptotic standard errors for estimating equation estimators, also known as exactly identified generalized method of moments estimators, are standard in the literature; see [Newey \(1984\)](#), [Newey and McFadden \(1994\)](#), [Tsiatis \(2006\)](#), and [Wooldridge \(2010\)](#). These standard errors always have a robust structure and have been generalized to cluster-robust standard errors (see [Wooldridge \[2010\]](#)).

The score equations and the functional form for the predicted mean survival time depend on the model for survival-time outcome  $F$ . We provide these details below, under [Functional-form details](#).

## Weighted-adjusted-censoring assumptions

All estimators that permit you to model the time to censoring are subject to three assumptions:

1. The censoring time must be random.
2. The censoring time must be from a known distribution.
3. The distribution of the censoring time cannot vary by treatment level.

We call these three requirements the WAC assumptions. If the WAC assumptions are violated, you can use either an RA estimator or the LAC version of the IPWRA estimator.

### □ Technical note

We now describe how the observed survival-time outcome  $t$  is generated from the random censoring time  $t_c$ , the received treatment  $\tau$ , and the potential-outcome survival times  $t_0$  and  $t_1$  under the WAC assumptions. First, each potential outcome is either censored or not censored.

$$\begin{aligned}\tilde{t}_0 &= t_c(t_0 \geq t_c) + t_0\{1 - (t_0 \geq t_c)\} \\ \tilde{t}_1 &= t_c(t_1 \geq t_c) + t_1\{1 - (t_1 \geq t_c)\}\end{aligned}$$

Under the WAC assumptions,  $t_c$  is a random variable from a known distribution, and  $t_c$  does not vary by treatment level.

Next, the received treatment  $\tau \in \{0, 1\}$  determines which, possibly censored, potential outcome is observed.

$$t = (1 - \tau)\tilde{t}_0 + \tau\tilde{t}_1$$

The 0/1 indicator for whether the observed  $t$  was censored, denoted by  $c$ , is given by

$$c = (1 - \tau)(t_0 \geq t_c) + \tau(t_1 \geq t_c)$$

□

## Weighted regression-adjusted estimators

As is standard in the survival literature, the RA estimators account for censored survival times by adding a term to the log-likelihood function for censored observations [see (1)]. In contrast, weighted regression-adjustment (WRA) estimators use weights to account for censored observations and are subject to the [WAC assumptions](#).

Wooldridge (2007) and Lin (2000) derived estimators for the regression parameters that maximize a weighted objective function of the uncensored observations. Each observation-level weight is the inverse of the probability of not being censored. Like the RA estimators, the WRA estimators use averages of the predicted mean survival times to estimate treatment-effect parameters.

The WRA estimators have the following logic.

- WRA1. Estimate by ML the parameters  $\gamma$  of a parametric survival-time model for the time to censoring  $t_c$ , in which  $F_c(t_c|\mathbf{w}, \gamma)$  is the distribution of  $t_c$  conditional on covariates  $\mathbf{w}$ . Note that the censoring process does not vary by treatment level and that we only observe  $t_c$  when the observed potential outcome was censored. Denote the estimates of  $\gamma$  by  $\hat{\gamma}$ .
- WRA2. For each treatment level  $\tau \in \{0, 1\}$ , estimate by weighted maximum likelihood (WML) the  $\beta_\tau$  parameters of a parametric survival-time model, denoted by  $F(t|\mathbf{x}, \tau, \beta_\tau)$ , where  $t$  is the survival-time outcome and  $\mathbf{x}$  are the covariates. The weights are the inverse of the estimated probabilities of not being censored,  $1/\{1 - F_c(t_c|\mathbf{w}, \hat{\gamma})\}$ , and only the uncensored observations are used. Denote the estimates of  $\beta_\tau$  by  $\hat{\beta}_{\text{wra}, \tau}$ .
- WRA3. Use the estimated  $\hat{\beta}_{\text{wra}, \tau}$  and the functional form implied by  $F(t|\mathbf{x}, \tau, \beta_\tau)$  to estimate the mean survival time, conditional on  $\mathbf{x}$  and treatment level  $\tau$ , for each sample observation, denoted by  $\hat{E}(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{wra}, \tau})$ . Conditional independence of the treatment and the survival-time potential outcomes ensures that  $E(t|\mathbf{x}, \tau, \beta_\tau) = E(t_\tau|\mathbf{x}, \beta_\tau)$ , where  $t_\tau$  is the potential survival-time outcome corresponding to treatment level  $\tau$ . Under correct model specification, sample averages of  $\hat{E}(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{wra}, \tau})$  consistently estimate the POM for treatment level  $\tau$ , denoted by  $\text{POM}_\tau$ .
- WRA4. A contrast of the estimated POMs estimates the ATE.

If estimating an ATET, step WRA3 is modified to use only the treated observations when estimating the POMs. A contrast of these POMs then estimates the ATET.

The contribution of the  $i$ th observation to the log likelihood that is maximized in step WRA1 is

$$L_{c, \text{wra}}(t_i, \mathbf{w}_i, \hat{\gamma}) = \varpi_i [c_i \ln\{f_c(t_i|\mathbf{w}_i, \hat{\gamma})\} + (1 - c_i) \ln\{1 - F_c(t_i|\mathbf{w}_i, \hat{\gamma})\}] \quad (4)$$

where  $\varpi_i$  is the observation-level weight,  $c_i$  is the 0/1 indicator for whether the survival-time observation on person  $i$  was censored,  $t_i$  is the observed failure time, and  $f_c(t_i|\mathbf{w}_i, \hat{\gamma})$  is the density corresponding to conditional time-to-censoring distribution  $F_c(t_i|\mathbf{w}_i, \hat{\gamma})$ . When  $c_i = 1$ ,  $t_i$  is the time to censoring. When  $c_i = 0$ , the censoring time is not observed; we only know that it is greater than the observed  $t_i$ . The first term accounts for the observations in which  $t_i$  is observed to be the censoring time, and the second term accounts for the observations in which the censoring time is greater than the observed  $t_i$ .

The contribution of the  $i$ th observation to the log likelihood that is maximized in step WRA2 is

$$L_{\text{wra}}(t_i, \mathbf{x}_i, \tau, \hat{\beta}_{\text{wra}, \tau}) = \varpi_i (\tau_i == \tau) \left[ \frac{(1 - c_i)}{\{1 - F_c(t_i|\mathbf{w}_i, \hat{\gamma})\}} \right] \ln\{f(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{wra}, \tau})\} \quad (5)$$

where  $f(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{wra}, \tau})$  is the density corresponding to distribution  $F(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{wra}, \tau})$ . Equation (5) does not contain a term that adjusts for censoring; see (1) for a comparison. Rather, it uses inverse-probability weights to account for both the censored and the uncensored observations.

The WRA estimators for the POMs simultaneously solve estimating equations (6a) through (6e) for  $\hat{\gamma}$ ,  $\hat{\beta}_{\text{wra}, 0}$ ,  $\hat{\beta}_{\text{wra}, 1}$ ,  $\widehat{\text{POM}}_{\text{wra}, 0}$ , and  $\widehat{\text{POM}}_{\text{wra}, 1}$ .

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{wra}}(t_i, \mathbf{w}_i, \hat{\gamma}, F_c) = \mathbf{0} \quad (6a)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{wra}}(t_i, \mathbf{x}_i, 0, \hat{\beta}_{\text{wra},0}, F) = \mathbf{0} \quad (6b)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{wra}}(t_i, \mathbf{x}_i, 1, \hat{\beta}_{\text{wra},1}, F) = \mathbf{0} \quad (6c)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \hat{E}(t_i | \mathbf{x}_i, \tau = 0, \hat{\beta}_{\text{wra},0}) - \widehat{\text{POM}}_{\text{wra},0} \right\} = 0 \quad (6d)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \hat{E}(t_i | \mathbf{x}_i, \tau = 1, \hat{\beta}_{\text{wra},1}) - \widehat{\text{POM}}_{\text{wra},1} \right\} = 0 \quad (6e)$$

where

$\mathbf{s}_{\text{wra}}(t_i, \mathbf{w}_i, \hat{\gamma}, F_c) = \frac{\partial L_{c,\text{wra}}(t_i, \mathbf{w}_i, \hat{\gamma})}{\partial \hat{\gamma}}$  is the vector of score equations from the ML estimator for  $\hat{\gamma}$  based on survival-time model  $F_c$ ,

$\mathbf{s}_{\text{wra}}(t_i, \mathbf{x}_i, 0, \hat{\beta}_{\text{wra},0}, F) = \frac{\partial L(t_i, \mathbf{x}_i, 0, \hat{\beta}_{\text{wra},0})}{\partial \hat{\beta}_{\text{wra},0}}$  is the vector of score equations from the WML estimator for  $\hat{\beta}_{\text{wra},0}$  based on survival-time model  $F$ ,

$\mathbf{s}_{\text{wra}}(t_i, \mathbf{x}_i, 1, \hat{\beta}_{\text{wra},1}, F) = \frac{\partial L(t_i, \mathbf{x}_i, 1, \hat{\beta}_{\text{wra},1})}{\partial \hat{\beta}_{\text{wra},1}}$  is the vector of score equations from the WML estimator for  $\hat{\beta}_{\text{wra},1}$  based on survival-time model  $F$ ,

$\hat{E}(t_i | \mathbf{x}_i, \tau = 0, \hat{\beta}_{\text{wra},0})$  is the predicted mean survival time assuming treatment level 0 for observation  $i$  conditional on  $\mathbf{x}_i$ , and

$\hat{E}(t_i | \mathbf{x}_i, \tau = 1, \hat{\beta}_{\text{wra},1})$  is the predicted mean survival time assuming treatment level 1 for observation  $i$  conditional on  $\mathbf{x}_i$ .

The observation-level scores  $\mathbf{s}_{\text{wra}}(t_i, \mathbf{x}_i, 0, \hat{\beta}_{\text{wra},0}, F)$  and  $\mathbf{s}_{\text{wra}}(t_i, \mathbf{x}_i, 1, \hat{\beta}_{\text{wra},1}, F)$  also depend on  $c_i$ ,  $\mathbf{w}_i$ ,  $\hat{\gamma}$ , and  $F_c$ , but we ignored this dependence to simplify the notation.

The ATE is estimated by replacing (6e) with

$$1/N \sum_{i=1}^N \varpi_i \left\{ \hat{E}(t_i | \mathbf{x}_i, \tau = 1, \hat{\beta}_{\text{wra},1}) - \widehat{\text{POM}}_{\text{wra},0} - \widehat{\text{ATE}}_{\text{wra}} \right\} = 0 \quad (7)$$

and the ATET is estimated by replacing (6e) and (7) with

$$1/N_1 \sum_{i=1}^N \varpi_i (\tau_i == 1) \left\{ \hat{E}(t_i | \mathbf{x}_i, \tau = 0, \hat{\beta}_{\text{wra},0}) - \widehat{\text{POM}}_{\text{wra},\text{cot},0} \right\} = 0$$

$$1/N_1 \sum_{i=1}^N \varpi_i (\tau_i == 1) \left\{ \hat{E}(t_i | \mathbf{x}_i, \tau = 1, \hat{\beta}_{\text{wra},1}) - \widehat{\text{POM}}_{\text{wra},\text{cot},0} - \widehat{\text{ATET}}_{\text{wra}} \right\} = 0$$

where  $\widehat{\text{POM}}_{\text{wra},\text{cot},0}$  is the estimated conditional-on-treatment POM.

## Inverse-probability-weighted estimators

IPW estimators are weighted averages of the observed outcome. The weights correct for missing data due to unobserved potential outcomes and censoring. Each weight is the inverse of the probability that a given value is observed. Observed values that were not likely to be observed have higher weights.

When the outcome variable is never censored, the missing data are the unobserved potential outcome and an observation's weight is the inverse of a treatment probability. When the outcome may be censored, the censoring is an additional source of missing data. In this case, an observation's weight is the inverse of the joint probability that an observation is uncensored and has a particular treatment level.

To define this joint probability, the censoring time must be random. In practice, we make the **WAC assumptions**.

As is standard in the survival-time literature, we assume that the censoring-time process is independent of treatment assignment after conditioning on the covariates. This conditional independence assumption implies that the probability that observation  $i$  receives treatment level 1 and is not censored is the product of the probability that  $i$  gets treatment level 1 and the probability that  $i$  is not censored at time  $t_i$ , which we denote by

$$p(\mathbf{z}_i, \boldsymbol{\alpha})\{1 - F_c(t_i|\mathbf{w}_i, \boldsymbol{\gamma})\}$$

where

$p(\mathbf{z}_i, \boldsymbol{\alpha})$  is the modeled probability that  $i$  gets treatment level 1, conditional on covariates  $\mathbf{z}_i$  with parameters  $\boldsymbol{\alpha}$ , and

$F_c(t_i|\mathbf{w}_i, \boldsymbol{\gamma})$  is the survival-time model for the censoring time, conditional on covariates  $\mathbf{w}_i$  with parameters  $\boldsymbol{\gamma}$ , and evaluated at time  $t_i$ .

Bai, Tsiatis, and O'Brien (2013) formally derive these weights to control jointly for the missing potential outcome and censoring.

The IPW estimators have the following logic.

- IPW1. Estimate by ML the parameters  $\boldsymbol{\gamma}$  of a parametric survival-time model for the time to censoring, in which  $F_c(t_c|\mathbf{w}, \boldsymbol{\gamma})$  is the distribution of censoring time, conditional on covariates  $\mathbf{w}$ . Denote the estimates of  $\boldsymbol{\gamma}$  by  $\hat{\boldsymbol{\gamma}}$ .
- IPW2. Estimate by ML the parameters  $\boldsymbol{\alpha}$  of a parametric model for the probability of treatment model  $p(\mathbf{z}_i, \boldsymbol{\alpha})$ . Denote the estimates of  $\boldsymbol{\alpha}$  by  $\hat{\boldsymbol{\alpha}}$ .
- IPW3. Use the  $\hat{\boldsymbol{\gamma}}$  estimated in IPW1 and the  $\hat{\boldsymbol{\alpha}}$  estimated in IPW2 to construct inverse-probability weights by (8a) for treatment level 1 and by (8b) for treatment level 0.

$$\omega_{i,1} = \frac{(\tau_i == 1)(c_i == 0)}{[p(\mathbf{z}_i, \hat{\boldsymbol{\alpha}})\{1 - F_c(t_i|\mathbf{w}_i, \hat{\boldsymbol{\gamma}})\}]} \tag{8a}$$

$$\omega_{i,0} = \frac{(\tau_i == 0)(c_i == 0)}{[\{1 - p(\mathbf{z}_i, \hat{\boldsymbol{\alpha}})\}\{1 - F_c(t_i|\mathbf{w}_i, \hat{\boldsymbol{\gamma}})\}]} \tag{8b}$$

- IPW4. Use the estimated weights to estimate each  $\text{POM}_\tau$  by a weighted average of the uncensored observations on the observed potential outcome.

The contribution of the  $i$ th observation to the log likelihood that is maximized in **step IPW1** is

$$L_{c,ipw}(t_i, \mathbf{w}_i, \hat{\gamma}) = \varpi_i [c_i \ln\{f_c(t_i|\mathbf{w}_i, \hat{\gamma})\} + (1 - c_i) \ln\{1 - F_c(t_i|\mathbf{w}_i, \hat{\gamma})\}]$$

where the definitions and intuition are as described after (4).

The contribution of the  $i$ th observation to the log likelihood that is maximized in **step IPW2** is

$$L_{p,ipw}(\tau_i, \mathbf{z}_i, \hat{\alpha}) = \varpi_i [(\tau_i == 1) \ln\{p(\mathbf{z}_i, \hat{\alpha})\} + \{1 - (\tau_i == 1)\} \ln\{1 - p(\mathbf{z}_i, \hat{\alpha})\}]$$

where  $p(\mathbf{z}_i, \hat{\alpha})$  is the model for the probability that  $i$  gets treatment level 1.

The IPW estimators for the POMs simultaneously solve estimating equations (9a) through (9d) for  $\hat{\gamma}$ ,  $\hat{\alpha}$ ,  $\widehat{\text{POM}}_{ipw,0}$ , and  $\widehat{\text{POM}}_{ipw,1}$ .

$$1/N \sum_{i=1}^N \mathbf{s}_{ipw}(t_i, \mathbf{w}_i, \hat{\gamma}, F_c) = \mathbf{0} \quad (9a)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{ipw}(\tau_i, \mathbf{z}_i, \hat{\alpha}, p) = \mathbf{0} \quad (9b)$$

$$1/N \sum_{i=1}^N \varpi_i \omega_{i,0} (t_i - \widehat{\text{POM}}_{ipw,0}) = 0 \quad (9c)$$

$$1/N \sum_{i=1}^N \varpi_i \omega_{i,1} (t_i - \widehat{\text{POM}}_{ipw,1}) = 0 \quad (9d)$$

where

$\mathbf{s}_{ipw}(t_i, \mathbf{w}_i, \hat{\gamma}, F_c) = \frac{\partial L_{c,ipw}(t_i, \mathbf{w}_i, \hat{\gamma})}{\partial \hat{\gamma}}$  is the vector of score equations from the ML estimator for  $\hat{\gamma}$  based on survival-time model  $F_c$ , and

$\mathbf{s}_{ipw}(\tau_i, \mathbf{z}_i, \hat{\alpha}, p) = \frac{\partial L_{p,ipw}(\tau_i, \mathbf{z}_i, \hat{\alpha})}{\partial \hat{\alpha}}$  is the vector of score equations from the ML estimator for  $\hat{\alpha}$  based on probability model  $p$ .

The literature on IPW estimators discusses using normalized versus unnormalized weights, with normalized weights doing better in simulation studies; see [Busso, DiNardo, and McCrary \(2014\)](#) for example. The way that weights enter moment equations (9c) and (9d) implies that they are normalized, because the scale of the weights does not affect the estimates.

The estimated ATE is computed as

$$\widehat{\text{POM}}_{ipw,1} - \widehat{\text{POM}}_{ipw,0} = \widehat{\text{ATE}}_{ipw}$$

The estimated ATET uses weights

$$\omega_{i,cot,1} = \frac{(\tau_i == 1)(c_i == 0)}{[\{1 - F_c(t_i|\mathbf{w}_i, \hat{\gamma})\}]} \quad (10a)$$

for treatment level 1 and

$$\omega_{i,cot,0} = \frac{p(\mathbf{z}_i, \hat{\alpha})(\tau_i == 0)(c_i == 0)}{[\{1 - p(\mathbf{z}_i, \hat{\alpha})\}\{1 - F_c(t_i|\mathbf{w}_i, \hat{\gamma})\}]} \quad (10b)$$



for treatment level 0, and replaces (9c) and (9d) with

$$1/N_1 \sum_{i=1}^N \varpi_i \omega_{i,\text{cot},0} (t_i - \widehat{\text{POM}}_{\text{ipw},\text{cot},0}) = 0 \quad (11a)$$

$$1/N_1 \sum_{i=1}^N \varpi_i \omega_{i,\text{cot},1} (t_i - \widehat{\text{POM}}_{\text{ipw},\text{cot},1}) = 0 \quad (11b)$$

and then computes

$$\widehat{\text{POM}}_{\text{ipw},\text{cot},1} - \widehat{\text{POM}}_{\text{ipw},\text{cot},0} = \widehat{\text{ATE}}_{\text{ipw}}$$

These IPW estimators can be viewed as weighted IPW estimators and are thus related to those in Hirano, Imbens, and Ridder (2003).

### Uncensored data

As mentioned, when the outcome variable is never censored, the missing data are the unobserved potential outcome and an observation's weight is the inverse of a treatment probability. In the never-censored case, the IPW estimators are identical to those implemented in `teffects ipw`; see *IPW estimators* under *Methods and formulas* in [TE] `teffects aipw`.

`stteffects ipw` computes the estimator for never-censored data when a censoring model is not specified and there are no censored observations in the sample. In the never-censored case, the following changes are made to the IPW estimator for the POMs and the ATE.

1. Step IPW1 is not performed.
2. The weights in (8a) and (8b) for the POMs and the ATE are replaced with (12a) for treatment level 1 and (12b) for treatment level 0.

$$\omega_{i,1} = \frac{(\tau_i == 1)}{p(\mathbf{z}_i, \widehat{\boldsymbol{\alpha}})} \quad (12a)$$

$$\omega_{i,0} = \frac{(\tau_i == 0)}{\{1 - p(\mathbf{z}_i, \widehat{\boldsymbol{\alpha}})\}} \quad (12b)$$

3. Only moment conditions (9b), (9c), and (9d) are used.

The following changes also are made to the IPW estimator for the ATET.

1. Step IPW1 is not performed.
2. The weights in (10a) and (10b) are replaced with (13a) for treatment level 1 and (13b) for treatment level 0.

$$\omega_{i,\text{cot},1} = (\tau_i == 1) \quad (13a)$$

$$\omega_{i,\text{cot},0} = \frac{p(\mathbf{z}_i, \widehat{\boldsymbol{\alpha}})(\tau_i == 0)}{\{1 - p(\mathbf{z}_i, \widehat{\boldsymbol{\alpha}})\}} \quad (13b)$$

3. Only moment conditions (9b), (11a), and (11b) are used.

## Inverse-probability-weighted regression-adjustment estimators

IPWRA estimators are averages of treatment-specific predicted conditional means that were made using missingness-adjusted regression parameters. These estimators are Wooldridge’s IPWRA for survival-time outcomes; see [Wooldridge \(2010, chap. 21\)](#) and [Wooldridge \(2007\)](#).

The censored observations can be handled either by weighting under the WAC assumptions to obtain the WAC-IPWRA estimator or by adding a term to the log-likelihood function (which we call likelihood-adjusted censoring) to obtain the LAC-IPWRA estimator. Correspondingly, there are two versions of formulas for the IPWRA estimator.

1. When a censoring model is specified, `stteffects ipwra` uses the formulas for the WAC-IPWRA estimator given in [Weighted-adjusted-censoring IPWRA](#).
2. When a censoring model is not specified, `stteffects ipwra` uses the formulas for the LAC-IPWRA given in [Likelihood-adjusted-censoring IPWRA](#), below.

The WAC-IPWRA estimator requires that some observations be censored and that the WAC assumptions hold; see [Weighted-adjusted-censoring assumptions](#), above. The LAC-IPWRA estimator handles the case in which no observations are censored and requires the weaker independent censoring assumptions, which allows for fixed censoring times.

## Weighted-adjusted-censoring IPWRA

When a censoring model is specified, `stteffects ipwra` uses the formulas for the WAC-IPWRA estimator to obtain the model-based weights that account for censoring. For notational conciseness and to reinforce its dependence on random censoring, we denote the WAC-IPWRA estimator by IPWRAR in lists and formulas. The WAC-IPWRA estimators have the following logic.

- IPWRAR1. Estimate by ML the parameters  $\gamma$  of a parametric survival-time model for the time to censoring, in which  $F_c(t_c|\mathbf{w}, \gamma)$  is the censoring-time distribution, conditional on covariates  $\mathbf{w}$ . We denote the estimates of  $\gamma$  by  $\hat{\gamma}$ .
- IPWRAR2. Estimate by ML the parameters  $\alpha$  of a parametric model for the probability of treatment model  $p(\mathbf{z}_i, \alpha)$ . We denote the estimates of  $\alpha$  by  $\hat{\alpha}$ .
- IPWRAR3. For each treatment level  $\tau \in \{0, 1\}$ , estimate by WML the parameters  $\beta_\tau$  of a parametric model for the survival-time outcome  $t$ , in which  $F(t|\mathbf{x}, \tau, \beta_\tau)$  is the distribution of  $t$  conditional on covariates  $\mathbf{x}$  and treatment level  $\tau$ . For the ATE, the weights are those in equations (8a) and (8b). For the ATET, the weights are those in equations (10a) and (10b). We denote the estimates of  $\beta_{\text{ipwrar}, \tau}$  by  $\hat{\beta}_\tau$ .
- IPWRAR4. Use the estimated  $\hat{\beta}_{\text{ipwrar}, \tau}$  and the functional form implied by  $F(t|\mathbf{x}, \tau, \beta_\tau)$  to estimate the mean survival time, conditional on  $\mathbf{x}$  and treatment level  $\tau$ , for each sample observation, denoted by  $\hat{E}(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{ipwrar}, \tau})$ . Conditional independence of the treatment and the survival-time potential outcomes ensures that  $E(t|\mathbf{x}, \tau, \beta_\tau) = E(t_\tau|\mathbf{x}, \beta_\tau)$ , where  $t_\tau$  is the potential survival-time outcome corresponding to treatment level  $\tau$ . Under correct model specification, sample averages of  $\hat{E}(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{ipwrar}, \tau})$  consistently estimate the POM for treatment level  $\tau$ , denoted by  $\text{POM}_\tau$ .

The contribution of the  $i$ th observation to the log likelihood that is maximized in [step IPWRAR1](#) is

$$L_{c, \text{ipwrar}}(t_i, \mathbf{w}_i, \hat{\gamma}) = \varpi_i [c_i \ln\{f_c(t_i|\mathbf{w}_i, \hat{\gamma})\} + (1 - c_i) \ln\{1 - F_c(t_i|\mathbf{w}_i, \hat{\gamma})\}]$$

where the definitions and intuition are as described after (4).

The contribution of the  $i$ th observation to the log likelihood that is maximized in [step IPWRAR2](#) is

$$L_{p,\text{ipwrar}}(\tau_i, \mathbf{z}_i, \hat{\boldsymbol{\alpha}}) = \varpi_i \{ (\tau_i == 1) \ln\{p(\mathbf{z}_i, \hat{\boldsymbol{\alpha}})\} + \{1 - (\tau_i == 1)\} \ln\{1 - p(\mathbf{z}_i, \hat{\boldsymbol{\alpha}})\} \}$$

where  $p(\mathbf{z}_i, \hat{\boldsymbol{\alpha}})$  is the model for the probability that  $i$  gets treatment level 1.

The weights and the parameters in [step IPWRAR3](#) used to estimate the ATE differ from those used to estimate the ATET. For the ATE, the contribution of the  $i$ th observation to the log likelihood that is maximized in [step IPWRAR3](#) is

$$L_{\text{ipwrar}}(t_i, \mathbf{x}_i, \tau, \hat{\boldsymbol{\beta}}_{\text{ipwrar,ate},\tau}) = \varpi_i \omega_{i,\tau} \ln\{f(t_i|\mathbf{x}_i, \tau, \hat{\boldsymbol{\beta}}_{\text{ipwrar,ate},\tau})\}$$

where  $\omega_{i,1}$  is given in [\(8a\)](#),  $\omega_{i,0}$  is given in [\(8b\)](#), and  $f(t_i|\mathbf{x}_i, \tau, \hat{\boldsymbol{\beta}}_{\text{ipwrar,ate},\tau})$  is the density corresponding to distribution  $F(t_i|\mathbf{x}_i, \tau, \hat{\boldsymbol{\beta}}_{\text{ipwrar,ate},\tau})$ . Like WRA, only the uncensored observations are used because the weights account for censoring.

The IPWRAR estimators for the POMs simultaneously solve estimating equations [\(14a\)](#) through [\(14f\)](#) for  $\hat{\boldsymbol{\gamma}}$ ,  $\hat{\boldsymbol{\alpha}}$ ,  $\hat{\boldsymbol{\beta}}_{\text{ipwrar,ate},0}$ ,  $\hat{\boldsymbol{\beta}}_{\text{ipwrar,ate},1}$ ,  $\widehat{\text{POM}}_{\text{ipwrar},0}$ , and  $\widehat{\text{POM}}_{\text{ipwrar},1}$ .

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwrar}}(t_i, \mathbf{w}_i, \hat{\boldsymbol{\gamma}}, F_c) = \mathbf{0} \quad (14a)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwrar}}(\tau_i, \mathbf{z}_i, \hat{\boldsymbol{\alpha}}, p) = \mathbf{0} \quad (14b)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwrar}}(t_i, \mathbf{x}_i, 0, \hat{\boldsymbol{\beta}}_{\text{ipwrar,ate},0}, F) = \mathbf{0} \quad (14c)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwrar}}(t_i, \mathbf{x}_i, 1, \hat{\boldsymbol{\beta}}_{\text{ipwrar,ate},1}, F) = \mathbf{0} \quad (14d)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \hat{E}(t_i|\mathbf{x}_i, \tau = 0, \hat{\boldsymbol{\beta}}_{\text{ipwrar,ate},0}) - \widehat{\text{POM}}_{\text{ipwrar},0} \right\} = 0 \quad (14e)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \hat{E}(t_i|\mathbf{x}_i, \tau = 1, \hat{\boldsymbol{\beta}}_{\text{ipwrar,ate},1}) - \widehat{\text{POM}}_{\text{ipwrar},1} \right\} = 0 \quad (14f)$$

where

$\mathbf{s}_{\text{ipwrar}}(t_i, \mathbf{w}_i, \hat{\boldsymbol{\gamma}}, F_c) = \frac{\partial L_{c,\text{ipwrar}}(t_i, \mathbf{w}_i, \hat{\boldsymbol{\gamma}})}{\partial \hat{\boldsymbol{\gamma}}}$  is the vector of score equations from the ML estimator for  $\hat{\boldsymbol{\gamma}}$  based on survival-time model  $F_c$ ,

$\mathbf{s}_{\text{ipwrar}}(\tau_i, \mathbf{z}_i, \hat{\boldsymbol{\alpha}}, p) = \frac{\partial L_{p,\text{ipwrar}}(\tau_i, \mathbf{z}_i, \hat{\boldsymbol{\alpha}})}{\partial \hat{\boldsymbol{\alpha}}}$  is the vector of score equations from the ML estimator for  $\hat{\boldsymbol{\alpha}}$  based on probability model  $p$ ,

$\mathbf{s}_{\text{ipwrar}}(t_i, \mathbf{x}_i, 0, \hat{\boldsymbol{\beta}}_{\text{ipwrar,ate},0}, F) = \frac{\partial L_{\text{ipwrar}}(t_i, \mathbf{x}_i, 0, \hat{\boldsymbol{\beta}}_{\text{ipwrar,ate},0})}{\partial \hat{\boldsymbol{\beta}}_{\text{ipwrar,ate},0}}$  is the vector of score equations from the ML estimator for  $\hat{\boldsymbol{\beta}}_{\text{ipwrar,ate},0}$  based on survival-time model  $F$ ,

$\mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwrrar,ate},1}, F) = \frac{\partial L_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwrrar,ate},1})}{\partial \widehat{\beta}_{\text{ipwrrar,ate},1}}$  is the vector of score equations

from the ML estimator for  $\widehat{\beta}_{\text{ipwrrar,ate},1}$  based on survival-time model  $F$ ,

$\widehat{E}(t_i | \mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ipwrrar,ate},0})$  is the predicted mean survival time assuming treatment level 0 for observation  $i$  conditional on  $\mathbf{x}_i$ , and

$\widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwrrar,ate},1})$  is the predicted mean survival time assuming treatment level 1 for observation  $i$  conditional on  $\mathbf{x}_i$ .

The ATE is estimated by replacing (14f) with

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwrrar,ate},1}) - \widehat{\text{POM}}_{\text{ipwrrar},0} - \widehat{\text{ATE}}_{\text{ipwrrar}} \right\} = 0$$

For the ATET, the contribution of the  $i$ th observation to the weighted log likelihood that is maximized in step IPWRRAR3 is

$$L_{\text{ipwrrar}}(t_i, \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwrrar,ate},\tau}) = \varpi_i \omega_{i,\text{cot},\tau}(\tau_i = \tau) \ln \{ f(t_i | \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwrrar,ate},\tau}) \}$$

where  $\omega_{i,\text{cot},1}$  is given in (10a),  $\omega_{i,\text{cot},0}$  is given in (10b), and  $f(t_i | \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwrrar,ate},\tau})$  is the density corresponding to distribution  $F(t_i | \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwrrar,ate},\tau})$ .

The WAC-IPWRA estimators for the conditional-on-treatment POMs simultaneously solve estimating equations (15a) through (15f) for  $\widehat{\beta}_{\text{ipwrrar,ate},0}$ ,  $\widehat{\beta}_{\text{ipwrrar,ate},1}$ ,  $\widehat{\gamma}$ ,  $\widehat{\alpha}$ ,  $\widehat{\text{POM}}_{\text{ipwrrar,cot},0}$ , and  $\widehat{\text{POM}}_{\text{ipwrrar,cot},1}$ .

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{w}_i, \widehat{\gamma}, F_c) = \mathbf{0} \quad (15a)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwrrar}}(\tau_i, \mathbf{z}_i, \widehat{\alpha}, p) = \mathbf{0} \quad (15b)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ipwrrar,ate},0}, F) = \mathbf{0} \quad (15c)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwrrar,ate},1}, F) = \mathbf{0} \quad (15d)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ipwrrar,ate},0}) - \widehat{\text{POM}}_{\text{ipwrrar,cot},0} \right\} = 0 \quad (15e)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwrrar,ate},1}) - \widehat{\text{POM}}_{\text{ipwrrar,cot},1} \right\} = 0 \quad (15f)$$

where

$\mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{w}_i, \widehat{\gamma}, F_c) = \frac{\partial L_{c,\text{ipwrrar}}(t_i, \mathbf{w}_i, \widehat{\gamma})}{\partial \widehat{\gamma}}$  is the vector of score equations from the ML estimator for  $\widehat{\gamma}$  based on survival-time model  $F_c$ ,

$\mathbf{s}_{\text{ipwrrar}}(\tau_i, \mathbf{z}_i, \widehat{\boldsymbol{\alpha}}, p) = \frac{\partial L_{p, \text{ipwrrar}}(\tau_i, \mathbf{z}_i, \widehat{\boldsymbol{\alpha}})}{\partial \widehat{\boldsymbol{\alpha}}}$  is the vector of score equations from the ML estimator for  $\widehat{\boldsymbol{\alpha}}$  based on probability model  $p$ ,

$\mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 0, \widehat{\boldsymbol{\beta}}_{\text{ipwrrar, atet, 0}}, F) = \frac{\partial L_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 0, \widehat{\boldsymbol{\beta}}_{\text{ipwrrar, atet, 0}})}{\partial \widehat{\boldsymbol{\beta}}_{\text{ipwrrar, atet, 0}}}$  is the vector of score equations from the WML estimator for  $\widehat{\boldsymbol{\beta}}_{\text{ipwrrar, atet, 0}}$  based on survival-time model  $F$ ,

$\mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 1, \widehat{\boldsymbol{\beta}}_{\text{ipwrrar, atet, 1}}, F) = \frac{\partial L_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 1, \widehat{\boldsymbol{\beta}}_{\text{ipwrrar, atet, 1}})}{\partial \widehat{\boldsymbol{\beta}}_{\text{ipwrrar, atet, 1}}}$  is the vector of score equations from the WML estimator for  $\widehat{\boldsymbol{\beta}}_{\text{ipwrrar, atet, 1}}$  based on survival-time model  $F$ ,

$\widehat{E}(t_i | \mathbf{x}_i, \tau = 0, \widehat{\boldsymbol{\beta}}_{\text{ipwrrar, atet, 0}})$  is the predicted mean survival time assuming treatment level 0 for observation  $i$  conditional on  $\mathbf{x}_i$ , and

$\widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\boldsymbol{\beta}}_{\text{ipwrrar, atet, 1}})$  is the predicted mean survival time assuming treatment level 1 for observation  $i$  conditional on  $\mathbf{x}_i$ .

The ATET is estimated by replacing (15f) with

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\boldsymbol{\beta}}_{\text{ipwrrar, atet, 1}}) - \widehat{\text{POM}}_{\text{ipwrrar, cot, 0}} - \widehat{\text{ATE}}_{\text{ipwrrar}} \right\} = 0$$

## Likelihood-adjusted-censoring IPWRA

When a censoring model is not specified, `stteffects ipwra` uses the formulas for the LAC-IPWRA estimator that add a term to the log-likelihood function. For notational conciseness and to reinforce its use of an additional term in the log likelihood, we denote the LAC-IPWRA estimator by `IPWRAL` in lists and formulas.

The methods and formulas for the LAC-IPWRA estimator differ in three ways from those for the WAC-IPWRA estimator.

1. No censoring model is specified, so LAC-IPWRA does not perform a version of `step IPWRAR1` and it does not use the moment equations (14a).
2. The weights only depend on the treatment level and treatment assignment probabilities, not on the censoring.
3. The WML estimator for  $\boldsymbol{\beta}_\tau$  includes a term for censored observations and censored observations are used. Recall that for the WAC-IPWRA estimator, the weights used in the WML estimator for  $\boldsymbol{\beta}_\tau$  account for the censoring, and the censored observations are not used in the WML estimator.

The LAC-IPWRA estimators have the following logic.

`IPWRAL1`. Estimate by ML the parameters  $\boldsymbol{\alpha}$  of a parametric model for the probability of treatment model  $p(\mathbf{z}_i, \boldsymbol{\alpha})$ .

`IPWRAL2`. For each treatment level  $\tau \in \{0, 1\}$ , estimate by WML the parameters  $\boldsymbol{\beta}_\tau$  of a parametric model for the survival-time outcome  $t$  in which  $F(t | \mathbf{x}, \tau, \boldsymbol{\beta}_\tau)$  is the distribution of  $t$  conditional on covariates  $\mathbf{x}$  and treatment level  $\tau$ . The weights depend only on the treatment level and the treatment-assignment probabilities. For the ATE, the weights are those in (12a) and (12b). For the ATET, the weights are those in (13a) and (13b). We denote the estimates of  $\boldsymbol{\beta}_\tau$  by  $\widehat{\boldsymbol{\beta}}_{\text{ipwrral}, \tau}$ .

IPWRAL3. Use the estimated  $\widehat{\beta}_{\text{ipwral},\tau}$  and the functional form implied by  $F(t|\mathbf{x},\tau,\beta_\tau)$  to estimate the mean survival time, conditional on  $\mathbf{x}$  and treatment level  $\tau$ , for each sample observation, denoted by  $\widehat{E}(t_i|\mathbf{x}_i,\tau,\widehat{\beta}_{\text{ipwral},\tau})$ . Conditional independence of the treatment and the survival-time potential outcomes ensures that  $E(t|\mathbf{x},\tau,\beta_\tau) = E(t_\tau|\mathbf{x},\beta_\tau)$ , where  $t_\tau$  is the potential survival-time outcome corresponding to treatment level  $\tau$ . Under correct model specification, sample averages of  $\widehat{E}(t_i|\mathbf{x}_i,\tau,\widehat{\beta}_{\text{ipwral},\tau})$  consistently estimate the POM for treatment level  $\tau$ , denoted by  $\text{POM}_\tau$ .

The contribution of the  $i$ th observation to the log likelihood that is maximized in [step IPWRAL1](#) is

$$L_{p,\text{ipwral}}(\tau_i, \mathbf{z}_i, \widehat{\alpha}) = \varpi_i \{ (\tau_i == 1) \ln\{p(\mathbf{z}_i, \widehat{\alpha})\} + \{1 - (\tau_i == 1)\} \ln\{1 - p(\mathbf{z}_i, \widehat{\alpha})\} \}$$

where  $p(\mathbf{z}_i, \widehat{\alpha})$  is the model for the probability that  $i$  gets treatment level 1.

The weights and the parameters in [step IPWRAL2](#) used to estimate the ATE differ from those used to estimate the ATET. For the ATE, the contribution of the  $i$ th observation to the log likelihood that is maximized in [step IPWRAL2](#) is

$$L_{\text{ipwral}}(t_i, \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,ate},\tau}) = (\tau_i == \tau) \varpi_i \omega_{i,\tau} \left\{ (1 - c_i) \ln\{f(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,ate},\tau})\} \right. \\ \left. c_i \ln\{1 - F(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,ate},\tau})\} \right\}$$

where  $\omega_{i,1}$  is given in [\(12a\)](#),  $\omega_{i,0}$  is given in [\(12b\)](#), and  $f(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,ate},\tau})$  is the density corresponding to distribution  $F(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,ate},\tau})$ . Unlike the WRA estimator, the censored observations are used, and there is a term in the likelihood function to account for censoring.

The LAC-IPWRA estimators for the POMs simultaneously solve estimating equations [\(16a\)](#) through [\(16e\)](#) for  $\widehat{\alpha}$ ,  $\widehat{\beta}_{\text{ipwral,ate},0}$ ,  $\widehat{\beta}_{\text{ipwral,ate},1}$ ,  $\widehat{\text{POM}}_{\text{ipwral},0}$ , and  $\widehat{\text{POM}}_{\text{ipwral},1}$ .

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwral}}(\tau_i, \mathbf{z}_i, \widehat{\alpha}, p) = \mathbf{0} \quad (16a)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwral}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ipwral,ate},0}, F) = \mathbf{0} \quad (16b)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwral}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwral,ate},1}, F) = \mathbf{0} \quad (16c)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i|\mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ipwral,ate},0}) - \widehat{\text{POM}}_{\text{ipwral},0} \right\} = 0 \quad (16d)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i|\mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwral,ate},1}) - \widehat{\text{POM}}_{\text{ipwral},1} \right\} = 0 \quad (16e)$$

where

$\mathbf{s}_{\text{ipwral}}(\tau_i, \mathbf{z}_i, \widehat{\alpha}, p) = \frac{\partial L_{p,\text{ipwral}}(\tau_i, \mathbf{z}_i, \widehat{\alpha})}{\partial \widehat{\alpha}}$  is the vector of score equations from the ML estimator for  $\widehat{\alpha}$  based on probability model  $p$ ,

$\mathbf{s}_{\text{ipwral}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ipwral,ate},0}, F) = \frac{\partial L_{\text{ipwral}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ipwral,ate},0})}{\partial \widehat{\beta}_{\text{ipwral,ate},0}}$  is the vector of score equations from the WML estimator for  $\widehat{\beta}_{\text{ipwral,ate},0}$  based on survival-time model  $F$ ,

$\mathbf{s}_{\text{ipwral}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwral,ate},1}, F) = \frac{\partial L_{\text{ipwral}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwral,ate},1})}{\partial \widehat{\beta}_{\text{ipwral,ate},1}}$  is the vector of score equations from the WML estimator for  $\widehat{\beta}_{\text{ipwral,ate},1}$  based on survival-time model  $F$ ,

$\widehat{E}(t_i | \mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ipwral,ate},0})$  is the predicted mean survival time assuming treatment level 0 for observation  $i$  conditional on  $\mathbf{x}_i$ , and

$\widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwral,ate},1})$  is the predicted mean survival time assuming treatment level 1 for observation  $i$  conditional on  $\mathbf{x}_i$ .

The ATE is estimated by replacing (16e) with

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwral,ate},1}) - \widehat{\text{POM}}_{\text{ipwral},0} - \widehat{\text{ATE}}_{\text{ipwral}} \right\} = 0$$

For the ATET, the contribution of the  $i$ th observation to the WML function that is maximized in `step IPWRAL2` is

$$L_{\text{ipwral}}(t_i, \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,atet},\tau}) = (\tau_i == \tau) \varpi_i \omega_{i,\text{cot},\tau} \left\{ (1 - c_i) \ln \{ f(t_i | \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,atet},\tau}) \} \right. \\ \left. c_i \ln \{ 1 - F(t_i | \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,atet},\tau}) \} \right\}$$

where  $\omega_{i,\text{cot},1}$  is given in (13a),  $\omega_{i,\text{cot},0}$  is given in (13b), and  $f(t_i | \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,atet},\tau})$  is the density corresponding to distribution  $F(t_i | \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,atet},\tau})$ . Again unlike the WRA, the censored observations are used, and there is a term in the likelihood function to account for censoring.

The LAC-IPWRA estimators for the conditional-on-treatment POMs simultaneously solve estimating equations (17a) through (17e) for  $\widehat{\alpha}$ ,  $\widehat{\beta}_{\text{ipwral,atet},0}$ ,  $\widehat{\beta}_{\text{ipwral,atet},1}$ ,  $\widehat{\text{POM}}_{\text{ipwral,cot},0}$ , and  $\widehat{\text{POM}}_{\text{ipwral,cot},1}$ .

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwral}}(\tau_i, \mathbf{z}_i, \widehat{\alpha}, p) = \mathbf{0} \quad (17a)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwral}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ipwral,atet},0}, F) = \mathbf{0} \quad (17b)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwral}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwral,atet},1}, F) = \mathbf{0} \quad (17c)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ipwral,atet},0}) - \widehat{\text{POM}}_{\text{ipwral,cot},0} \right\} = 0 \quad (17d)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwral,atet},1}) - \widehat{\text{POM}}_{\text{ipwral,cot},1} \right\} = 0 \quad (17e)$$

where

$\mathbf{s}_{\text{ipwral}}(\tau_i, \mathbf{z}_i, \hat{\boldsymbol{\alpha}}, p) = \frac{\partial L_{p,\text{ipwral}}(\tau_i, \mathbf{z}_i, \hat{\boldsymbol{\alpha}})}{\partial \hat{\boldsymbol{\alpha}}}$  is the vector of score equations from the ML estimator for  $\hat{\boldsymbol{\alpha}}$  based on probability model  $p$ ,

$\mathbf{s}_{\text{ipwral}}(t_i, \mathbf{x}_i, 0, \hat{\boldsymbol{\beta}}_{\text{ipwral,atet},0}, F) = \frac{\partial L_{\text{ipwral}}(t_i, \mathbf{x}_i, 0, \hat{\boldsymbol{\beta}}_{\text{ipwral,atet},0})}{\partial \hat{\boldsymbol{\beta}}_{\text{ipwral,atet},0}}$  is the vector of score equations from the WML estimator for  $\hat{\boldsymbol{\beta}}_{\text{ipwral,atet},0}$  based on survival-time model  $F$ ,

$\mathbf{s}_{\text{ipwral}}(t_i, \mathbf{x}_i, 1, \hat{\boldsymbol{\beta}}_{\text{ipwral,atet},1}, F) = \frac{\partial L_{\text{ipwral}}(t_i, \mathbf{x}_i, 1, \hat{\boldsymbol{\beta}}_{\text{ipwral,atet},1})}{\partial \hat{\boldsymbol{\beta}}_{\text{ipwral,atet},1}}$  is the vector of score equations from the WML estimator for  $\hat{\boldsymbol{\beta}}_{\text{ipwral,atet},1}$  based on survival-time model  $F$ ,

$\hat{E}(t_i | \mathbf{x}_i, \tau = 0, \hat{\boldsymbol{\beta}}_{\text{ipwral,atet},0})$  is the predicted mean survival time assuming treatment level 0 for observation  $i$  conditional on  $\mathbf{x}_i$ , and

$\hat{E}(t_i | \mathbf{x}_i, \tau = 1, \hat{\boldsymbol{\beta}}_{\text{ipwral,atet},1})$  is the predicted mean survival time assuming treatment level 1 for observation  $i$  conditional on  $\mathbf{x}_i$ .

The ATET is estimated by replacing (17e) with

$$1/N \sum_{i=1}^N \varpi_i \left\{ \hat{E}(t_i | \mathbf{x}_i, \tau = 1, \hat{\boldsymbol{\beta}}_{\text{ipwral,atet},1}) - \widehat{\text{POM}}_{\text{ipwral,cot},0} - \widehat{\text{ATET}}_{\text{ipwral}} \right\} = 0$$

## Functional-form details

In this section, we specify the functional forms for the conditional distribution function used in the survival-time outcome model  $F$ , the conditional distribution function used in the survival-time censoring model  $F_c$ , and the conditional distribution used to model the treatment probabilities  $p$ .

You may choose among the same set of conditional distribution functions for either  $F$  or  $F_c$ : exponential, weibull, lnormal, or gamma.

Name	Cumulative	Density	Mean
exponential	$1 - \exp(-\lambda_i t_i)$	$\lambda_i \exp(-\lambda_i t_i)$	$1/\lambda_i$
Weibull	$1 - \exp\{-(\lambda_i t_i)^{s_i}\}$	$s_i t_i^{s_i-1} \lambda_i^{s_i} \exp\{-(\lambda_i t_i)^{s_i}\}$	$(1/\lambda_i) \Gamma\{(s_i + 1)/s_i\}$
log normal	$\Phi\{(\ln(t_i) - \lambda_i)/s_i\}$	$(1/(s_i t_i)) \phi\{(\ln(t_i) - \lambda_i)/s_i\}$	$\exp(\lambda_i + s_i^2/2)$
gamma	$\text{gammap}\{s_i, (s_i t_i/\lambda_i)\}$	$(s_i^{s_i} t_i^{s_i-1}) / \{\lambda_i^{s_i} \Gamma(s_i)\} \exp(-s_i t_i/\lambda_i)$	$\lambda_i$

where the following table specifies how  $\lambda_i$  and  $s_i$  are parameterized in terms of the covariates  $\mathbf{x}_i$  and the ancillary covariates  $\tilde{\mathbf{x}}_i$ , respectively.

Name	$\lambda_i$	$s_i$
exponential	$\exp(-\mathbf{x}_i \boldsymbol{\beta})$	
Weibull	$\exp(-\mathbf{x}_i \boldsymbol{\beta})$	$\exp(\tilde{\mathbf{x}}_i \tilde{\boldsymbol{\beta}})$
log normal	$\mathbf{x}_i \boldsymbol{\beta}$	$\exp(\tilde{\mathbf{x}}_i \tilde{\boldsymbol{\beta}})$
gamma	$\exp(\mathbf{x}_i \boldsymbol{\beta})$	$\exp(-2\tilde{\mathbf{x}}_i \tilde{\boldsymbol{\beta}})$



For the treatment-assignment models, the `probit` model uses the standard normal distribution, the `logit` uses the standard logistic distribution, the `hetprobit` model uses

$$\Phi\{\mathbf{z}_1\alpha_1/\exp(\mathbf{z}_2\alpha_2)\}$$

and the multinomial logit uses

$$p(\mathbf{z}, t) = \exp(\mathbf{z}\alpha_t) / \left\{ 1 + \sum_{k=1}^q \exp(\mathbf{z}\alpha_k) \right\}$$

where the notation is defined below.

In the `hetprobit` model,  $\mathbf{z}_1$  are the covariates specified in the treatment-assignment specification,  $\mathbf{z}_2$  are the covariates specified in the `hetprobit()` option, and  $\alpha_1$  and  $\alpha_2$  are the corresponding coefficients.

In the multinomial logit model,  $\mathbf{z}$  are the covariates specified in the treatment-assignment specification and  $\alpha_k$  are the coefficients; see [R] `mlogit` for further details.

## References

- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.
- Bai, X., A. A. Tsiatis, and S. M. O'Brien. 2013. Doubly robust estimators of treatment-specific survival distributions in observational studies with stratified sampling. *Biometrics* 69: 830–839. <https://doi.org/10.1111/biom.12076>.
- Busso, M., J. DiNardo, and J. McCrary. 2014. New evidence on the finite sample properties of propensity score reweighting and matching estimators. *Review of Economics and Statistics* 96: 885–897. [https://doi.org/10.1162/REST\\_a\\_00431](https://doi.org/10.1162/REST_a_00431).
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.
- Drukker, D. M. 2014. Using gmm to solve two-step estimation problems. *The Stata Blog: Not Elsewhere Classified*. <http://blog.stata.com/2014/12/08/using-gmm-to-solve-two-step-estimation-problems/>.
- Guo, S., and M. W. Fraser. 2015. *Propensity Score Analysis: Statistical Methods and Applications*. 2nd ed. Thousand Oaks, CA: SAGE.
- Hirano, K., G. W. Imbens, and G. Ridder. 2003. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica* 71: 1161–1189. <https://doi.org/10.1111/1468-0262.00442>.
- Imbens, G. W. 2000. The role of the propensity score in estimating dose–response functions. *Biometrika* 87: 706–710. <https://doi.org/10.1093/biomet/87.3.706>.
- . 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics* 86: 4–29. <https://doi.org/10.1162/003465304323023651>.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.
- Lin, D. Y. 2000. Linear regression analysis of censored medical costs. *Biostatistics* 1: 35–47. <https://doi.org/10.1093/biostatistics/1.1.35>.
- Newey, W. K. 1984. A method of moments interpretation of sequential estimators. *Economics Letters* 14: 201–206. [https://doi.org/10.1016/0165-1765\(84\)90083-1](https://doi.org/10.1016/0165-1765(84)90083-1).

- Newey, W. K., and D. L. McFadden. 1994. Large sample estimation and hypothesis testing. In Vol. 4 of *Handbook of Econometrics*, ed. R. F. Engle and D. L. McFadden, 2111–2245. Amsterdam: Elsevier. [https://doi.org/10.1016/S1573-4412\(05\)80005-4](https://doi.org/10.1016/S1573-4412(05)80005-4).
- Robins, J. M., and A. Rotnitzky. 1995. Semiparametric efficiency in multivariate regression models with missing data. *Journal of the American Statistical Association* 90: 122–129. <https://doi.org/10.2307/2291135>.
- . 2006. Inverse probability weighting in survival analysis. In *Survival and Event History Analysis*, ed. N. Keiding and P. K. Andersen, 266–271. Chichester, UK: Wiley.
- Robins, J. M., A. Rotnitzky, and L. P. Zhao. 1995. Analysis of semiparametric regression models for repeated outcomes in the presence of missing data. *Journal of the American Statistical Association* 90: 106–121. <https://doi.org/10.1080/01621459.1995.10476493>.
- Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55. <https://doi.org/10.2307/2335942>.
- Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701. <https://doi.org/10.1037/h0037350>.
- Tsiatis, A. A. 2006. *Semiparametric Theory and Missing Data*. New York: Springer.
- Vittinghoff, E., D. V. Glidden, S. C. Shiboski, and C. E. McCulloch. 2012. *Regression Methods in Biostatistics: Linear, Logistic, Survival, and Repeated Measures Models*. 2nd ed. New York: Springer.
- Wooldridge, J. M. 2002. Inverse probability weighted M-estimators for sample selection, attrition, and stratification. *Portuguese Economic Journal* 1: 117–139. <https://doi.org/10.1007/s10258-002-0008-x>.
- . 2007. Inverse probability weighted estimation for general missing data problems. *Journal of Econometrics* 141: 1281–1301. <https://doi.org/10.1016/j.jeconom.2007.02.002>.
- . 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

## Also see

- [TE] **stteffects postestimation** — Postestimation tools for stteffects
- [TE] **stteffects intro** — Introduction to treatment effects for observational survival-time data
- [ST] **streg** — Parametric survival models
- [ST] **stset** — Declare data to be survival-time data
- [U] **20 Estimation and postestimation commands**

## Postestimation commands

The following postestimation commands are of special interest after `stteffects`:

Command	Description
<code>teoverlap</code>	overlap plots
<code>tebalance</code>	check balance of covariates

The following standard postestimation commands are also available:

Command	Description
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
<code>hausman</code>	Hausman’s specification test
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of coefficients
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
<code>predict</code>	propensity scores, censored survival probability, etc.
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

# predict

## Description for predict

`predict` creates a new variable containing predictions such as treatment effects, conditional means, propensity scores, linear predictions, and log square roots of latent variances.

## Menu for predict

Statistics > Postestimation

## Syntaxes for predict

Syntaxes are presented under the following headings:

*Syntax for predict after stteffects ipw*  
*Syntax for predict after stteffects ipwra*  
*Syntax for predict after stteffects ra*  
*Syntax for predict after stteffects wra*

### Syntax for predict after stteffects ipw

```
predict [type] { stub* | newvar | newvarlist } [if] [in]
      [, statistic tlevel(treat_level)]
```

```
predict [type] stub* [if] [in], scores
```

<i>statistic</i>	Description
Main	
<code>ps</code>	propensity score; the default
<code>censurv</code>	censored survival probability
<code>xb</code>	linear prediction for propensity score
<code>cxb</code>	linear prediction for censoring model
<code>lnsigma</code>	log square root of latent variance (for treatment model <code>hetprobit()</code> )
<code>clnshape</code>	log of conditional latent shape (for censoring distribution Weibull, log normal, or gamma)

If you do not specify `tlevel()` and only specify one new variable, `ps` assumes `tlevel()` specifies the first treatment level.

If you do not specify `tlevel()` and only specify one new variable, `xb` and `lnsigma` assume `tlevel()` specifies the first noncontrol treatment level.

You specify one or  $t$  new variables with `ps`, where  $t$  is the number of treatment levels.

You specify one or  $t-1$  new variables with `xb` and `lnsigma`.

You specify one new variable with `censurv`, `cxb`, and `clnshape`.

## Syntax for predict after stteffects ipwra

```
predict [type] { stub* | newvar | newvarlist } [if] [in]
      [, statistic tlevel(treat_level)]
```

```
predict [type] stub* [if] [in], scores
```

<i>statistic</i>	Description
Main	
<code>te</code>	treatment effect; the default
<code>cmean</code>	conditional mean at treatment level
<code>ps</code>	propensity score
<code>censurv</code>	censored survival probability
<code>xb</code>	linear prediction for outcome model
<code>cxb</code>	linear prediction for censoring model
<code>psxb</code>	linear prediction for propensity score
<code>lnshape</code>	log of conditional latent shape (for outcome distribution Weibull, log normal, or gamma) at treatment level
<code>clnshape</code>	log of conditional latent shape (for censoring distribution Weibull, log normal, or gamma)
<code>pslnsigma</code>	log square root of latent variance (for treatment model <code>hetprobit()</code> ) for propensity score

If you do not specify `tlevel()` and only specify one new variable, `te` and `psxb` assume `tlevel()` specifies the first noncontrol treatment level.

If you do not specify `tlevel()` and only specify one new variable, `cmean`, `ps`, `xb`, and `pslnsigma` assume `tlevel()` specifies the first treatment level.

You specify one or  $t$  new variables with `cmean`, `ps`, `xb`, and `lnshape`, where  $t$  is the number of treatment levels.

You specify one or  $t-1$  new variables with `te`, `psxb`, and `pslnsigma`.

You specify one new variable with `censurv`, `cxb`, and `clnshape`.

**Syntax for predict after stteffects ra**

```
predict [type] { stub* | newvar | newvarlist } [if] [in]
      [, statistic tlevel(treat_level)]
```

```
predict [type] stub* [if] [in], scores
```

<i>statistic</i>	Description
Main	
<b>te</b>	treatment effect; the default
<b>cmean</b>	conditional mean at treatment level
<b>xb</b>	linear prediction for outcome model
<b>lnshape</b>	log of conditional latent shape (for outcome distribution Weibull, log normal, or gamma) at treatment level

If you do not specify `tlevel()` and only specify one new variable, `te` assumes `tlevel()` specifies the first noncontrol treatment level.

If you do not specify `tlevel()` and only specify one new variable, `cmean`, `xb`, and `lnshape` assume `tlevel()` specifies the first treatment level.

You specify one or  $t$  new variables with `cmean`, `xb`, and `lnshape`, where  $t$  is the number of treatment levels.

You specify one or  $t-1$  new variables with `te`.

**Syntax for predict after stteffects wra**

```
predict [type] { stub* | newvar | newvarlist } [if] [in]
      [, statistic tlevel(treat_level)]
```

```
predict [type] stub* [if] [in], scores
```

<i>statistic</i>	Description
Main	
<b>te</b>	treatment effect; the default
<b>cmean</b>	conditional mean at treatment level
<b>censurv</b>	censored survival probability
<b>xb</b>	linear prediction for outcome model
<b>cxb</b>	linear prediction for censoring model
<b>lnshape</b>	log of conditional latent shape (for outcome distribution Weibull, log normal, or gamma) at treatment level
<b>clnshape</b>	log of conditional latent shape (for censoring distribution Weibull, log normal, or gamma)

If you do not specify `tlevel()` and only specify one new variable, `te` assumes `tlevel()` specifies the first noncontrol treatment level.

If you do not specify `tlevel()` and only specify one new variable, `cmean`, `xb`, and `lnshape` assume `tlevel()` specifies the first treatment level.

You specify one or  $t$  new variables with `cmean`, `xb`, and `lnshape`, where  $t$  is the number of treatment levels.

You specify one or  $t-1$  new variables with `te`.

You specify one new variable with `censurv`, `cxb`, and `clnshape`.

## Options for predict

Options are presented under the following headings:

*Options for predict after stteffects ipw*  
*Options for predict after stteffects ipwra*  
*Options for predict after stteffects ra*  
*Options for predict after stteffects wra*

### Options for predict after stteffects ipw

Main

**ps**, the default, calculates the propensity score of each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

**censurv** calculates the survivor probability from the time-to-censoring model. (In other words, it calculates the probability that an outcome is not censored.) This option is allowed only if a censoring model is specified at estimation time. You need to specify only one new variable.

**xb** calculates the propensity score linear prediction at each noncontrol level of the treatment or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

**cxb** calculates the linear prediction of the censoring model. This option is allowed only if a censoring model is specified at estimation time. You need to specify only one new variable.

**lnsigma** calculates the log square root of the latent variance. This option is valid only when treatment model `hetprobit()` is used. You need to specify only one new variable.

**clnshape** calculates the log of the conditional latent shape parameter of the censoring distribution. This option is valid when censoring distribution Weibull, log normal, or gamma is used. You need to specify only one new variable.

**tlevel(*treat\_level*)** specifies the treatment level for prediction.

**scores** calculates the score variables. Parameter-level scores are computed for the treatment mean and average treatment-effect equations. Equation-level scores are computed for the censoring and propensity-score equations.

The  $j$ th new variable will contain the scores for the  $j$ th parameter in the coefficient table if  $j \leq t$ , where  $t$  is the number of treatment levels. Otherwise, it will contain the scores for fitted equation  $j - t$  following the first  $t$  parameters in the coefficient table.

### Options for predict after stteffects ipwra

Main

**te**, the default, calculates the treatment effect for each noncontrol treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

**cmean** calculates the conditional mean for each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

- ps** calculates the propensity score of each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.
- censurv** calculates the survivor probability from the time-to-censoring model. (In other words, it calculates the probability that an outcome is not censored.) This option is allowed only if a censoring model is specified at estimation time. You need to specify only one new variable.
- xb** calculates the outcome model linear prediction at each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.
- ccb** calculates the linear prediction of the censoring model. This option is allowed only if a censoring model is specified at estimation time. You need to specify only one new variable.
- psxb** calculates the propensity score linear prediction at each noncontrol level of the treatment or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).
- lnshape** calculates the log of the conditional latent shape parameter for each treatment level or the treatment level specified in `tlevel()`. This option is valid when outcome distribution Weibull, log normal, or gamma is used. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.
- clnshape** calculates the log of the conditional latent shape parameter for the censoring distribution. This option is valid when censoring distribution Weibull, log normal, or gamma is used. You need to specify only one new variable.
- pslnsigma** calculates the log square root of the latent variance for the propensity score. This option is valid only when treatment model `hetprobit()` is used. You need to specify only one new variable.
- tlevel(*treat\_level*)** specifies the treatment level for prediction.
- scores** calculates the score variables. Parameter-level scores are computed for the treatment mean and average treatment-effect equations. Equation-level scores are computed for the outcome, censoring, and propensity-score equations.
- The  $j$ th new variable will contain the scores for the  $j$ th parameter in the coefficient table if  $j \leq t$ , where  $t$  is the number of treatment levels. Otherwise, it will contain the scores for fitted equation  $j - t$  following the first  $t$  parameters in the coefficient table.

## Options for predict after stteffects ra

### Main

- te**, the default, calculates the treatment effect for each noncontrol treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).
- cmean** calculates the conditional mean for each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.
- xb** calculates the outcome model linear prediction at each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.



`lnshape` calculates the log of the conditional latent shape parameter for each treatment level or the treatment level specified in `tlevel()`. This option is valid when the outcome distribution Weibull, log normal, or gamma is used. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`tlevel(treat_level)` specifies the treatment level for prediction.

`scores` calculates the score variables. Parameter-level scores are computed for the treatment mean and average treatment-effect equations. Equation-level scores are computed for the outcome equations.

The  $j$ th new variable will contain the scores for the  $j$ th parameter in the coefficient table if  $j \leq t$ , where  $t$  is the number of treatment levels. Otherwise, it will contain the scores for fitted equation  $j - t$  following the first  $t$  parameters in the coefficient table.

## Options for predict after stteffects wra

### Main

`te`, the default, calculates the treatment effect for each noncontrol treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

`cmean` calculates the conditional mean for each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`censurv` calculates the survivor probability from the time-to-censoring model. (In other words, it calculates the probability that an outcome is not censored.) This option is allowed only if a censoring model is specified at estimation time. You need to specify only one new variable.

`xb` calculates the outcome model linear prediction at each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`lnshape` calculates the log of the conditional latent shape parameter for each treatment level or the treatment level specified in `tlevel()`. This option is valid when the outcome distribution Weibull, log normal, or gamma is used. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`clnshape` calculates the log of the conditional latent shape parameter of the censoring distribution. This option is valid when the censoring distribution Weibull, log normal, or gamma is used. You need to specify only one new variable.

`tlevel(treat_level)` specifies the treatment level for prediction.

`scores` calculates the score variables. Parameter-level scores are computed for the treatment mean and average treatment-effect equations. Equation-level scores are computed for the outcome and censoring equations.

The  $j$ th new variable will contain the scores for the  $j$ th parameter in the coefficient table if  $j \leq t$ , where  $t$  is the number of treatment levels. Otherwise, it will contain the scores for fitted equation  $j - t$  following the first  $t$  parameters in the coefficient table.

## Remarks and examples

Checking model specification is the most frequent reason for postestimation computation after `stteffects.teoverlap` provides a graphical method for checking the overlap assumption; see [TE] [teoverlap](#). Summarizing the estimated probabilities provides another check. Recall that the reciprocals of these estimated probabilities are used as weights by some of the estimators. If the estimated probabilities are too small, the weights get too large and the estimators become unstable.

We estimate the average treatment effect of smoking on the time to a second heart attack by inverse-probability weighting; see [example 1](#) of [TE] [stteffects ipw](#) for background.

```
. use https://www.stata-press.com/data/r17/sheart
(Time to second heart attack (fictional))
. stteffects ipw (smoke age exercise education) (age exercise diet education)

      Failure _d: fail
      Analysis time _t: atime
Iteration 0:   EE criterion = 2.042e-18
Iteration 1:   EE criterion = 5.191e-31
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: logit
Censoring model: Weibull
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE smoke (Smoker vs Nonsmoker)	-2.22226	.6307573	-3.52	0.000	-3.458522	-.9859983
POmean smoke Nonsmoker	4.235569	.5210937	8.13	0.000	3.214244	5.256894

Below, we compute the estimated probabilities of being a Nonsmoker and store them in `ps0`. Likewise, the estimated probabilities of being a Smoker are stored in `ps1`.

```
. predict ps0 ps1, ps
```

The overlap condition requires that each of these probabilities be sufficiently greater than 0 and less than 1 for every individual; see [Assumptions and trade-offs](#) under [Remarks and examples](#) in [TE] [stteffects intro](#).

In practice, we know that weighting estimators perform poorly when the weights become too large. This approach requires that the probability of being a Nonsmoker not be too small among Nonsmokers and that the probability of being a Smoker not be too small among Smokers. Below, we summarize these probabilities.

```
. summarize ps0 if fail==1 & smoke==0
```

Variable	Obs	Mean	Std. dev.	Min	Max
ps0	716	.6712529	.138754	.3872543	.9840293

```
. summarize ps1 if fail==1 & smoke==1
```

Variable	Obs	Mean	Std. dev.	Min	Max
ps1	492	.4101277	.1101277	.0850604	.6125538

The minimum probability of being a Nonsmoker among Nonsmokers is 0.39. The minimum probability of being a Smoker among Smokers is 0.09. Neither minimum seems too small.

Estimating survival-time treatment effects also uses weights to adjust for censored outcomes; see [TE] [stteffects intro](#). Thus we require that the probability of an uncensored failure also be sufficiently greater than 0. Below, we compute the estimated probabilities of failure and summarize them among those that fail.

```
. predict fprob2, censurv
. summarize fprob if fail==1
```

Variable	Obs	Mean	Std. dev.	Min	Max
fprob2	1,208	.7246067	.2143543	.0364246	.9999086

The minimum probability of 0.04 does not appear too small.

□ Technical note

The previous discussion builds on the intuition that the weights used in a weighting estimator should not be too large.

This technical note goes a little further by explicitly computing the weights and using them to replicate the inverse-probability-weighted point estimate for the Nonsmoker potential-outcome mean.

We now compute the weights using the predicted probabilities computed in the examples above and then use `mean` to compute the weighted average that estimates the potential-outcome mean for Nonsmokers.

```
. generate double ipw0 = 1/(ps0*fprob)
. mean _t [pw=ipw0] if smoke==0 & fail==1
```

Mean estimation Number of obs = 716

	Mean	Std. err.	[95% conf. interval]	
_t	4.235569	.5820212	3.092894	5.378244

The weights account for data lost to the Smoker potential outcome or to censoring by increasing the importance of observations that were observed to be Nonsmoker failure times even though they were not likely to be observed.

The point estimate matches that reported by `stteffects ipw`; the standard errors differ because `mean` takes the estimated weights as given. See [Inverse-probability-weighted estimators](#) under *Methods and formulas* in [TE] [stteffects ipwra](#).



## References

- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.
- Bai, X., A. A. Tsiatis, and S. M. O'Brien. 2013. Doubly robust estimators of treatment-specific survival distributions in observational studies with stratified sampling. *Biometrics* 69: 830–839. <https://doi.org/10.1111/biom.12076>.
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.
- Guo, S., and M. W. Fraser. 2015. *Propensity Score Analysis: Statistical Methods and Applications*. 2nd ed. Thousand Oaks, CA: SAGE.
- Imbens, G. W. 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics* 86: 4–29. <https://doi.org/10.1162/003465304323023651>.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.
- Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55. <https://doi.org/10.2307/2335942>.
- Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701. <https://doi.org/10.1037/h0037350>.
- Tsiatis, A. A. 2006. *Semiparametric Theory and Missing Data*. New York: Springer.
- Vittinghoff, E., D. V. Glidden, S. C. Shiboski, and C. E. McCulloch. 2012. *Regression Methods in Biostatistics: Linear, Logistic, Survival, and Repeated Measures Models*. 2nd ed. New York: Springer.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

## Also see

- [TE] **tebalance** — Check balance after teffects or stteffects estimation
- [TE] **teoverlap** — Overlap plots
- [TE] **stteffects ipw** — Survival-time inverse-probability weighting
- [TE] **stteffects ipwra** — Survival-time inverse-probability-weighted regression adjustment
- [TE] **stteffects ra** — Survival-time regression adjustment
- [TE] **stteffects wra** — Survival-time weighted regression adjustment
- [U] **20 Estimation and postestimation commands**

# Title

**stteffects ra** — Survival-time regression adjustment

<a href="#">Description</a>	<a href="#">Quick start</a>	<a href="#">Menu</a>	<a href="#">Syntax</a>
<a href="#">Options</a>	<a href="#">Remarks and examples</a>	<a href="#">Stored results</a>	<a href="#">Methods and formulas</a>
<a href="#">References</a>	<a href="#">Also see</a>		

## Description

`stteffects ra` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational survival-time data by regression adjustment (RA). RA uses averages of treatment-specific predicted mean survival times to estimate mean survival times for each potential outcome. Contrasts of these predicted mean survival times estimate the treatment effects. `stteffects ra` offers several choices for the model used to predict mean survival time. Binary and multivalued treatments are accommodated.

See [\[TE\] stteffects intro](#) for an overview of estimating treatment effects from observational survival-time data.

## Quick start

Specify `time` as observed failure time and `fail` as failure indicator

```
stset time, failure(fail)
```

ATE from a Weibull model for `time` on `x1` and `x2` with binary treatment `treat2`

```
stteffects ra (x1 x2) (treat2)
```

As above, but estimate the ATET

```
stteffects ra (x1 x2) (treat2), atet
```

As above, but estimate the potential-outcome means

```
stteffects ra (x1 x2) (treat2), pomeans
```

ATE of `treat2` using a gamma model for `time`

```
stteffects ra (x1 x2, gamma) (treat2)
```

ATE for each level of three-valued treatment `treat3`

```
stteffects ra (x1 x2) (treat3)
```

As above, and specify that `treat3 = 3` is the control level using the value label “MyControl” for 3

```
stteffects ra (x1 x2) (treat3), control("MyControl")
```

## Menu

Statistics > Treatment effects > Survival outcomes > Regression adjustment

## Syntax

```
stteffects ra (omvarlist [, omoptions]) (tvar) [if] [in] [, stat options]
```

*omvarlist* specifies the variables that predict the survival-time variable in the outcome model.

*tvar* must contain integer values representing the treatment levels.

<i>omoptions</i>	Description
Model	
<code>weibull</code>	Weibull; the default
<code>exponential</code>	exponential
<code>gamma</code>	two-parameter gamma
<code>lnormal</code>	lognormal
<code>ancillary</code> ( <i>avarlist</i> [, <code>noconstant</code> ])	specify variables used to model ancillary parameter
<code>noconstant</code>	suppress constant from outcome model

<i>stat</i>	Description
Stat	
<code>ate</code>	estimate average treatment effect in population; the default
<code>atet</code>	estimate average treatment effect on the treated
<code>pomeans</code>	estimate potential-outcome means

<i>options</i>	Description
SE/Robust	
<code>vce</code> ( <i>vcetype</i> )	<i>vcetype</i> may be <code>robust</code> , <code>cluster <i>clustvar</i></code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level</code> (#)	set confidence level; default is <code>level(95)</code>
<code>aequations</code>	display auxiliary-equation results
<code>noshow</code>	do not show st setting information
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
<code>iterinit</code> (#)	specify starting-value iterations; seldom used
Advanced	
<code>control</code> (#   <i>label</i> )	specify the level of <i>tvar</i> that is the control
<code>tlevel</code> (#   <i>label</i> )	specify the level of <i>tvar</i> that is the treatment
<code>coeflegend</code>	display legend instead of statistics

You must `stset` your data before using `stteffects`; see [ST] [stset](#).

`omvarlist` and `avarlist` may contain factor variables; see [U] [11.4.3 Factor variables](#).

`bootstrap`, `by`, `collect`, `jackknife`, and `statsby` are allowed; see [U] [11.1.10 Prefix commands](#).

Weights are not allowed with the `bootstrap` prefix; see [R] [bootstrap](#).

`fweights`, `iwweights`, and `pweights` may be specified using `stset`; see [Weights](#) under *Remarks and examples* in [ST] [stset](#). However, weights may not be specified if you are using the `bootstrap` prefix.

`coeflegend` does not appear in the dialog box.

See [U] [20 Estimation and postestimation commands](#) for more capabilities of estimation commands.

## Options

### Model

`ancillary`(*avarlist* [, `noconstant` ]) specifies the variables used to model the ancillary parameter.

By default, the ancillary parameter does not depend on covariates. Specifying `ancillary`(*avarlist*, `noconstant`) causes the constant to be suppressed in the model for the ancillary parameter.

`noconstant`; see [R] [Estimation options](#).

### Stat

*stat* is one of three statistics: `ate`, `atet`, or `pomeans`. `ate` is the default.

`ate` specifies that the average treatment effect be estimated.

`atet` specifies that the average treatment effect on the treated be estimated.

`pomeans` specifies that the potential-outcome means for each treatment level be estimated.

### SE/Robust

`vce`(*vcetype*) specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster` *clustvar*), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce\\_option](#).

### Reporting

`level`(#); see [R] [Estimation options](#).

`aequations` specifies that the results for the outcome-model or treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

`noshow` prevents `stteffects ra` from showing the key `st` variables. This option is rarely used because most people type `stset`, `show` or `stset`, `noshow` to permanently set whether they want to see these variables mentioned at the top of the output of every `st` command; see [ST] [stset](#).

*display\_options*: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap`(#), `fvwrapon`(*style*), `cformat`(%*fmt*), `pformat`(%*fmt*), `sformat`(%*fmt*), and `nolstretch`; see [R] [Estimation options](#).

### Maximization

*maximize\_options*: `iterate`(#), `[no]log`, and `from`(*init\_specs*); see [R] [Maximize](#). These options are seldom used.

*init\_specs* is one of

*matname* [, `skip copy`]

# [, # ...], `copy`

`iterinit(#)` specifies the maximum number of iterations used to calculate the starting values. This option is seldom used.

---

Advanced

---

`control(# | label)` specifies the level of *tvar* that is the control. The default is the first treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with the statistic `pomeans`. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(# | label)` specifies the level of *tvar* that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

The following option is available with `stteffects` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

## Remarks and examples

If you are not familiar with the framework for treatment-effects estimation from observational survival-time data, please see [\[TE\] stteffects intro](#).

RA estimators use contrasts of the averages of treatment-specific predicted mean outcomes to estimate treatment effects. RA estimators use a two-step approach to estimating treatment effects:

1. For each treatment level, fit a model of the survival-time outcome on the same set of covariates.
2. Compute the averages of the predicted outcomes for each subject within each treatment level.

These averages estimate the potential-outcome means (POMs). Contrasts of these averages estimate the ATES. By restricting the computations of the averages to the subset of treated subjects, we obtain estimates of the ATETs.

Here we note only a few entry points to the vast literature on RA estimators. [Imbens \(2004\)](#), [Imbens and Wooldridge \(2009\)](#), [Cameron and Trivedi \(2005, chap. 25\)](#), [Wooldridge \(2010, chap. 21\)](#), and [Vittinghoff et al. \(2012, chap. 9\)](#) provide excellent general introductions to estimating ATES and to RA estimators in particular.

Like `streg` and other survival-time commands, `stteffects ra` uses the outcome variable and the failure indicator computed by, and optionally weights specified with, `stset`. `stteffects ra` is not appropriate for data with time-varying covariates, also known as multiple-record survival-time data, or for delayed-entry data.

### ▷ Example 1: Estimating the ATE

Suppose we wish to study the effect of smoking on the time to a second heart attack among women aged 45–55 years. In our fictional `sheart` dataset, `atime` is the observed time in years to a second heart attack or censoring, and `fail` is the 0/1 indicator that a second heart attack was observed. (When `fail` is 1, `atime` records the time to the second heart attack; when `fail` is 0, `atime` records a censored observation of the time to a second heart attack.) We previously `stset` these data; see [A quick tour of the estimators](#) in [\[TE\] stteffects intro](#).



The treatment, smoking, is stored in the 0/1 indicator `smoke`. These data also contain age at the time of the first heart attack (`age`), and indices of the level of exercise (`exercise`), diet quality (`diet`), and education (`education`) prior to the first heart attack.

We can use `stteffects ra` to estimate the ATE by RA. We model the mean survival time using the default Weibull model, controlling for age, exercise, diet, and education, and we specify that `smoke` is the treatment variable.

```
. use https://www.stata-press.com/data/r17/sheart
(Time to second heart attack (fictional))
. stteffects ra (age exercise diet education) (smoke)
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:   EE criterion = 1.006e-14
Iteration 1:   EE criterion = 2.305e-25
Survival treatment-effects estimation          Number of obs      =      2,000
Estimator      : regression adjustment
Outcome model  : Weibull
Treatment model: none
Censoring model: none
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE smoke (Smoker vs Nonsmoker)	-1.956657	.3331787	-5.87	0.000	-2.609676	-1.303639
POmean smoke Nonsmoker	4.243974	.2620538	16.20	0.000	3.730358	4.75759

When every woman smoked in the population of women aged 45–55 years who have had a heart attack, the average time to a second heart attack is estimated to be 1.96 years less than when no women in the population of interest smoked. The estimated average time to a second heart attack when no women in the population of interest smoked is 4.24 years. In other words, if every woman in the population of interest smoked, then the average time to a second heart attack would fall by an estimated 46% relative to the case when no women smoked.



### ► Example 2: Changing the outcome model

Instead of a Weibull model for the outcome model, we could have used an exponential, a gamma, or a lognormal model. By way of comparison, we use a gamma model and the same covariates to estimate the ATE.

```

. stteffects ra (age exercise diet education, gamma) (smoke)
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 6.212e-25
Iteration 1:  EE criterion = 7.256e-32
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : regression adjustment
Outcome model  : gamma
Treatment model: none
Censoring model: none

```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
smoke (Smoker vs Nonsmoker)	-1.801787	.2924388	-6.16	0.000	-2.374956	-1.228617
POMean						
smoke Nonsmoker	3.994327	.2258257	17.69	0.000	3.551717	4.436937

The estimated ATE of  $-1.80$  and control-level POM of  $3.99$  are similar to those of  $-1.96$  and  $4.24$  obtained from the Weibull model in [example 1](#). The ratio of the estimated ATE to the control-level POM indicates a 45% reduction instead of the 46% reduction obtained from the Weibull model.

◀

### ► Example 3: Estimating the ratio of the ATE to the control-level POM

The ratio of the ATE to the control-level POM measures the importance of the effect. In [example 1](#), we computed the point estimate of this ratio from the output, but we were left without a confidence interval. In this example, we use `nlcom` to compute a point estimate and a confidence interval.

Below, we refit the model from [example 1](#), specifying the `coeflegend` option to learn the parameter names. We use the parameter names in `nlcom` to estimate the ratio of the ATE to the control-level POM.

```
. stteffects ra (age exercise diet education) (smoke), coeflegend
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 1.006e-14
Iteration 1:  EE criterion = 2.305e-25
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : regression adjustment
Outcome model  : Weibull
Treatment model: none
Censoring model: none
```

_t	Coefficient	Legend
ATE smoke (Smoker vs Nonsmoker)	-1.956657	_b[ATE:r1vs0.smoke]
POmean smoke Nonsmoker	4.243974	_b[POmean:0.smoke]

```
. nlcom _b[ATE:r1vs0.smoke] / _b[POmean:0.smoke]
      _nl_1:  _b[ATE:r1vs0.smoke] / _b[POmean:0.smoke]
```

_t	Coefficient	Std. err.	z	P> z	[95% conf. interval]
_nl_1	-.4610437	.0598709	-7.70	0.000	-.5783885    -.3436988

The output shows that when every woman smoked, the average time to a second heart attack falls by an estimated 46% relative to the case when no women smoked, as we computed earlier. We also obtain a 95% confidence interval of 34% to 58% for this estimate.



### ▷ Example 4: Estimating the ATET

Intuitively, the ATET measures the effect of the treatment on an at-risk subpopulation. Sometimes the subpopulation that gets the treatment defines such an at-risk subpopulation. The ATET has the added benefit that it can be estimated under weaker conditions than the ATE; see [Assumptions and trade-offs](#) in [TE] [stteffects intro](#).

```

. stteffects ra (age exercise diet education) (smoke), atet
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 1.006e-14
Iteration 1:  EE criterion = 2.970e-26
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : regression adjustment
Outcome model  : Weibull
Treatment model: none
Censoring model: none

```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATET						
smoke (Smoker vs Nonsmoker)	-1.527476	.2489203	-6.14	0.000	-2.015351	-1.039602
P0mean						
smoke Nonsmoker	3.436937	.2217808	15.50	0.000	3.002255	3.87162

When every woman in the subpopulation smoked, the average time to a second heart attack is estimated to be 1.53 years less than when no women in the subpopulation smoked. The estimated average time to a second heart attack when no women in the subpopulation smoked is 3.44 years. ◀

### ► Example 5: Fixed or random censoring time

The time to censoring in survival-time data can be random or deterministic, although it must be independent of treatment assignment and the potential outcomes; see [Kalbfleisch and Prentice \(2002, chap. 3\)](#) for the standard case and see [The correct adjustment for censoring assumption](#) under [Assumptions and trade-offs](#) in [TE] [stteffects intro](#) for the treatment-effects case.

The RA estimator and the likelihood-adjusted-censoring version of the inverse-probability-weighted RA estimator can accommodate a fixed time to censoring; see [The correct adjustment for censoring assumption](#) in [TE] [stteffects intro](#). (The estimators that handle censoring by weighting cannot accommodate a fixed time to censoring because the weights are not well defined with a fixed time to censoring.)

We have fictional data on the time to rearrest among men aged 25–35 who were previously in prison for a felony conviction (`ptime`). The time to censoring is fixed in these data because individuals were followed for a maximum of five years.

Some of the young men chose to enter a vocational training program before release from prison; `train` is 1 for participants and 0 for nonparticipants. The dataset also contains `fail` (which is 1 if the observed time is a failure time and 0 if it is time to censoring), `age` at the time of the first arrest (age), an index of the parents' socioeconomic level (`parental`), and the number of years behind in school at the time of the first arrest (`edeficit`).

We estimate the ATET because we wish to allow the gains from the training program to be related to an unobservable characteristic that affects who self-selects into the program; see [Average treatment effect on the treated](#) in [TE] [stteffects intro](#).

We model the outcome as a function of age, parental, and edeficit.

```
. use https://www.stata-press.com/data/r17/recid2, clear
(Time to rearrest (fictional))
. stteffects ra (age parental edeficit) (train), atet

Failure _d: fail
Analysis time _t: rtime
Iteration 0: EE criterion = 1.875e-23
Iteration 1: EE criterion = 4.785e-32

Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : regression adjustment
Outcome model  : Weibull
Treatment model: none
Censoring model: none
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATET						
train						
(Student						
vs						
Nonstudent)	2.440919	.4689057	5.21	0.000	1.52188	3.359957
P0mean						
train						
Nonstudent	2.062029	.1231492	16.74	0.000	1.820661	2.303397

When everyone who selected the training got the training, the average time to rearrest is 2.44 years later than the average rearrest time if none of those who selected the training got the training. The average rearrest time if none of those who selected the training got the training is 2.06 years. In other words, the average time to rearrest increases from about 2.06 years to about 4.50 years for the subpopulation of young men who self-selected into the prerelease vocational training program.



## Stored results

stteffects ra stores the following in e():

### Scalars

- e(N)                    number of observations
- e(nj)                  number of observations for treatment level *j*
- e(N\_clust)            number of clusters
- e(k\_eq)                number of equations in e(b)
- e(k\_levels)            number of levels in treatment variable
- e(treated)            level of treatment variable defined as treated
- e(control)            level of treatment variable defined as control
- e(converged)          1 if converged, 0 otherwise

### Macros

- e(cmd)                stteffects
- e(cmdline)            command as typed
- e(dead)                \_d
- e(depvar)             \_t
- e(tvar)                name of treatment variable
- e(subcmd)             ra
- e(omodel)            outcome model: weibull, exponential, gamma, or lognormal
- e(stat)                statistic estimated: ate, atet, or pomeans

<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

## Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

## Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

## Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -values, and confidence intervals
-----------------------	--

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

## Methods and formulas

The methods and formulas for the RA estimators implemented in `stteffects ra` are given in *Methods and formulas* of [TE] `stteffects ipwra`.

## References

- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.
- Bai, X., A. A. Tsiatis, and S. M. O'Brien. 2013. Doubly robust estimators of treatment-specific survival distributions in observational studies with stratified sampling. *Biometrics* 69: 830–839. <https://doi.org/10.1111/biom.12076>.
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.
- Guo, S., and M. W. Fraser. 2015. *Propensity Score Analysis: Statistical Methods and Applications*. 2nd ed. Thousand Oaks, CA: SAGE.
- Imbens, G. W. 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics* 86: 4–29. <https://doi.org/10.1162/003465304323023651>.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.
- Kalbfleisch, J. D., and R. L. Prentice. 2002. *The Statistical Analysis of Failure Time Data*. 2nd ed. New York: Wiley.
- Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55. <https://doi.org/10.2307/2335942>.

- Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701. <https://doi.org/10.1037/h0037350>.
- Tsiatis, A. A. 2006. *Semiparametric Theory and Missing Data*. New York: Springer.
- Vittinghoff, E., D. V. Glidden, S. C. Shiboski, and C. E. McCulloch. 2012. *Regression Methods in Biostatistics: Linear, Logistic, Survival, and Repeated Measures Models*. 2nd ed. New York: Springer.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

## Also see

- [TE] [stteffects postestimation](#) — Postestimation tools for stteffects
- [TE] [stteffects intro](#) — Introduction to treatment effects for observational survival-time data
- [ST] [streg](#) — Parametric survival models
- [ST] [stset](#) — Declare data to be survival-time data
- [U] [20 Estimation and postestimation commands](#)

# Title

**stteffects wra** — Survival-time weighted regression adjustment

<a href="#">Description</a>	<a href="#">Quick start</a>	<a href="#">Menu</a>	<a href="#">Syntax</a>
<a href="#">Options</a>	<a href="#">Remarks and examples</a>	<a href="#">Stored results</a>	<a href="#">Methods and formulas</a>
<a href="#">References</a>	<a href="#">Also see</a>		

## Description

`stteffects wra` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational survival-time data with random time to censoring. Estimation is by weighted regression adjustment (WRA). WRA estimators use inverse-probability-of-censoring adjusted regression coefficients to compute averages of treatment-level predicted outcomes. Contrasts of these averages estimate the treatment effects. WRA uses estimated weights from a time-to-censoring model to account for censored survival times instead of including a term in the likelihood function. `stteffects wra` offers several choices for the functional forms of the outcome model and the time-to-censoring model. Binary and multivalued treatments are accommodated.

See [TE] [stteffects intro](#) for an overview of estimating treatment effects from observational survival-time data.

## Quick start

Specify `time` as observed failure time and `fail` as failure indicator

```
stset time, failure(fail)
```

ATE from a Weibull model for `time` on `x1` and `x2` with binary treatment `treat2` and a Weibull model on `x1` and `x2` for censoring

```
stteffects wra (x1 x2) (treat2) (x1 x2)
```

As above, but estimate the ATET

```
stteffects wra (x1 x2) (treat2) (x1 x2), atet
```

ATE of `treat2` using a gamma model for `time` and a gamma censoring model

```
stteffects wra (x1 x2, gamma) (treat2) (x1 x2, gamma)
```

ATE for each level of three-valued treatment `treat3`

```
stteffects wra (x1 x2) (treat3) (x1 x2)
```

As above, and specify that `treat3 = 3` is the control level using the value label “MyControl” for 3

```
stteffects wra (x1 x2) (treat3) (x1 x2), control("MyControl")
```

## Menu

Statistics > Treatment effects > Survival outcomes > Weighted regression adjustment



## Syntax

```
stteffects wra (omvarlist [, omoptions]) (tvar) (cmvarlist [, cmoptions])
  [if] [in] [, stat options]
```

*omvarlist* specifies the variables that predict the survival-time variable in the outcome model.

*tvar* must contain integer values representing the treatment levels.

*cmvarlist* specifies the variables that predict censoring in the censoring model.

<i>omoptions</i>	Description
Model	
<u>weibull</u>	Weibull; the default
<u>exponential</u>	exponential
<u>gamma</u>	two-parameter gamma
<u>lnormal</u>	lognormal
<u>ancillary</u> ( <i>avarlist</i> [, <u>noconstant</u> ])	specify variables used to model ancillary parameter
<u>noconstant</u>	suppress constant from outcome model

<i>cmoptions</i>	Description
Model	
<u>weibull</u>	Weibull; the default
<u>exponential</u>	exponential
<u>gamma</u>	two-parameter gamma
<u>lnormal</u>	lognormal
<u>ancillary</u> ( <i>avarlist</i> [, <u>noconstant</u> ])	specify variables used to model ancillary parameter
<u>noconstant</u>	suppress constant from censoring model

<i>stat</i>	Description
Stat	
<u>ate</u>	estimate average treatment effect in population; the default
<u>atet</u>	estimate average treatment effect on the treated
<u>pomeans</u>	estimate potential-outcome means

<i>options</i>	Description
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>robust</code> , <code>cluster <i>clustvar</i></code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>aequations</code>	display auxiliary-equation results
<code>noshow</code>	do not show st setting information
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
<code>iterinit(#)</code>	specify starting-value iterations; seldom used
Advanced	
<code>pstolerance(#)</code>	set tolerance for the overlap assumption
<code>osample(<i>newvar</i>)</code>	identify observations that violate the overlap assumption
<code>control(#   <i>label</i>)</code>	specify the level of <i>tvar</i> that is the control
<code>tlevel(#   <i>label</i>)</code>	specify the level of <i>tvar</i> that is the treatment
<code>coeflegend</code>	display legend instead of statistics

You must `stset` your data before using `stteffects`; see [ST] `stset`.

`omvarlist`, `cmvarlist`, and `avarlist` may contain factor variables; see [U] 11.4.3 **Factor variables**.

`bootstrap`, `by`, `collect`, `jackknife`, and `statsby` are allowed; see [U] 11.1.10 **Prefix commands**.

Weights are not allowed with the `bootstrap` prefix; see [R] `bootstrap`.

`fweights`, `iweights`, and `pweights` may be specified using `stset`; see *Weights* under *Remarks and examples* in [ST] `stset`. However, weights may not be specified if you are using the `bootstrap` prefix.

`coeflegend` does not appear in the dialog box.

See [U] 20 **Estimation and postestimation commands** for more capabilities of estimation commands.

## Options

Model

`ancillary(avarlist [, noconstant ])` specifies the variables used to model the ancillary parameter.

By default, the ancillary parameter does not depend on covariates. Specifying `ancillary(avarlist, noconstant)` causes the constant to be suppressed in the model for the ancillary parameter.

`ancillary()` may be specified for the model for survival-time outcome, for the model for the censoring variable, or for both. If `ancillary()` is specified for both, the varlist used for each model may be different.

`noconstant`; see [R] **Estimation options**.

## Stat

*stat* is one of three statistics: *ate*, *atet*, or *pomeans*. *ate* is the default.

*ate* specifies that the average treatment effect be estimated.

*atet* specifies that the average treatment effect on the treated be estimated.

*pomeans* specifies that the potential-outcome means for each treatment level be estimated.

## SE/Robust

*vce*(*vcetype*) specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (*robust*), that allow for intragroup correlation (*cluster clustvar*), and that use bootstrap or jackknife methods (*bootstrap*, *jackknife*); see [R] *vce\_option*.

## Reporting

*level*(#); see [R] **Estimation options**.

*aequations* specifies that the results for the outcome-model or treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

*noshow* prevents *stteffects wra* from showing the key *st* variables. This option is rarely used because most people type *stset*, *show* or *stset*, *noshow* to permanently set whether they want to see these variables mentioned at the top of the output of every *st* command; see [ST] **stset**.

*display\_options*: *noci*, *nopvalues*, *noomitted*, *vsquish*, *noemptycells*, *baselevels*, *allbaselevels*, *nofvlabel*, *fvwrap*(#), *fvwrapon*(*style*), *cformat*(%*fmt*), *pformat*(%*fmt*), *sformat*(%*fmt*), and *nolstretch*; see [R] **Estimation options**.

## Maximization

*maximize\_options*: *iterate*(#), [no]log, and *from*(*init\_specs*); see [R] **Maximize**. These options are seldom used.

*init\_specs* is one of

*matname* [ , skip copy ]

# [ , # ... ], copy

*iterinit*(#) specifies the maximum number of iterations used to calculate the starting values. This option is seldom used.

## Advanced

*pstolerance*(#) specifies the tolerance used to check the overlap assumption. The default value is *pstolerance*(1e-5). *stteffects* will exit with an error if an observation has an estimated propensity score smaller than that specified by *pstolerance*().

*osample*(*newvar*) specifies that indicator variable *newvar* be created to identify observations that violate the overlap assumption.

*control*(# | *label*) specifies the level of *tvar* that is the control. The default is the first treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. *control*() may not be specified with the statistic *pomeans*. *control*() and *tlevel*() may not specify the same treatment level.

`tlevel(#|label)` specifies the level of *tvar* that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

The following option is available with `stteffects` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

## Remarks and examples

If you are not familiar with the framework for treatment-effects estimation from observational survival-time data, please see [TE] [stteffects intro](#).

Weighted regression-adjustment (WRA) estimators use estimated weights to account for censoring when estimating outcome-regression parameters. The estimated outcome-regression parameters are used to compute averages of treatment-level predicted outcomes. Contrasts of these averages estimate the treatment effects.

WRA estimators use a three-step approach to estimating treatment effects:

1. They estimate the parameters of a time-to-censoring model and compute inverse-probability-of-censoring weights.
2. Using the estimated inverse-probability-of-censoring weights, they use weighted maximum likelihood estimators for the outcome for each treatment level and obtain the treatment-specific predicted mean outcomes for each subject. The inverse-probability-of-censoring weights account for right-censored survival times.
3. They compute the means of the treatment-specific predicted mean outcomes. Contrasts of these averages provide the estimates of the ATEs. By restricting the computations of the means to the subset of treated subjects, we can obtain the ATETs.

WRA estimators differ from RA estimators in that WRA estimators use weights to account for observations lost to censoring while RA estimators use an additional term in the likelihood function. A model for the time to censoring is used to estimate the weights.

WRA estimators require more assumptions than RA estimators. Specifically, they require that the censoring time be random and that the time-to-censoring model be well specified. The implemented WRA estimators also require that the time-to-censoring process not vary by treatment level. The RA estimator and the likelihood-adjusted-censoring version of the inverse-probability-weighted RA estimator do not require these extra assumptions, because they use a likelihood term instead of weights to adjust for the data lost to censoring; see [TE] [stteffects ra](#) and [TE] [stteffects ipwra](#).

Here we note only a few entry points to the vast literature on weighted estimators. [Imbens \(2004\)](#), [Imbens and Wooldridge \(2009\)](#), [Robins and Rotnitzky \(2006\)](#), [Wooldridge \(2002, 2007\)](#), [Cameron and Trivedi \(2005, chap. 25\)](#), [Wooldridge \(2010, chap. 21\)](#), and [Vittinghoff et al. \(2012, chap. 9\)](#) provide excellent general introductions to estimating ATEs and to WRA estimators in particular.

Like `streg` and other survival-time commands, `stteffects wra` uses the outcome variable and the failure indicator computed by, and optionally weights specified with, `stset`. `stteffects wra` is not appropriate for data with time-varying covariates, also known as multiple-record survival-time data, or for delayed-entry data.

▷ Example 1: Estimating the ATE

Suppose we wish to study the effect of smoking on the time to a second heart attack among women aged 45–55 years. In our fictional `sheart` dataset, `atime` is the observed time in years to a second heart attack or censoring, and `fail` is the 0/1 indicator that a second heart attack was observed. (When `fail` is 1, `atime` records the time to the second heart attack; when `fail` is 0, `atime` records a censored observation of the time to a second heart attack.) We previously `stset` these data; see [A quick tour of the estimators](#) in [TE] `stteffects intro`.

The treatment, smoking, is stored in the 0/1 indicator `smoke`. These data also contain age at the time of the first heart attack (`age`), and indices of the level of exercise (`exercise`), diet quality (`diet`), and education (`education`) prior to the first heart attack.

We can use `stteffects wra` to estimate the ATE by WRA. We model the mean survival time using the default Weibull outcome model with `age`, `exercise`, `diet`, and `education` as covariates, and we specify that `smoke` is the treatment variable. We also specify the default Weibull time-to-censoring model and include `age`, square of `age`, `exercise`, and `education`.

```
. use https://www.stata-press.com/data/r17/sheart
(Time to second heart attack (fictional))
. stteffects wra (age exercise diet education)
> (smoke)
> (age c.age#c.age exercise diet education)

      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 4.096e-18
Iteration 1:  EE criterion = 1.406e-29

Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : weighted regression adjustment
Outcome model  : Weibull
Treatment model: none
Censoring model: Weibull
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE smoke (Smoker vs Nonsmoker)	-2.374174	.6017498	-3.95	0.000	-3.553582	-1.194766
POmean smoke Nonsmoker	4.302131	.5528943	7.78	0.000	3.218478	5.385784

When every woman smoked in the population of women aged 45–55 years who have had a heart attack, the average time to a second heart attack is estimated to be 2.37 years less than when no women in the subpopulation of interest smoked. The estimated average time to a second heart attack when no women in the subpopulation of interest smoked is 4.30 years.

## Stored results

`stteffects wra` stores the following in `e()`:

### Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level $j$
<code>e(N_clust)</code>	number of clusters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(converged)</code>	1 if converged, 0 otherwise

### Macros

<code>e(cmd)</code>	<code>stteffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(dead)</code>	<code>_d</code>
<code>e(depvar)</code>	<code>_t</code>
<code>e(tvar)</code>	name of treatment variable
<code>e(subcmd)</code>	<code>wra</code>
<code>e(omodel)</code>	outcome model: weibull, exponential, gamma, or lognormal
<code>e(cmodel)</code>	censoring model: weibull, exponential, gamma, or lognormal
<code>e(stat)</code>	statistic estimated: ate, atet, or pomeans
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<i>vctype</i> specified in <code>vce()</code>
<code>e(vctype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

### Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

### Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

### Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, $p$ -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r`-class command is run after the estimation command.

## Methods and formulas

The methods and formulas for the WRA estimators implemented in `stteffects wra` are given in *Methods and formulas* of [TE] `stteffects ipwra`.

## References

- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.
- Bai, X., A. A. Tsiatis, and S. M. O'Brien. 2013. Doubly robust estimators of treatment-specific survival distributions in observational studies with stratified sampling. *Biometrics* 69: 830–839. <https://doi.org/10.1111/biom.12076>.
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.
- Guo, S., and M. W. Fraser. 2015. *Propensity Score Analysis: Statistical Methods and Applications*. 2nd ed. Thousand Oaks, CA: SAGE.
- Imbens, G. W. 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics* 86: 4–29. <https://doi.org/10.1162/003465304323023651>.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.
- Robins, J. M., and A. Rotnitzky. 2006. Inverse probability weighting in survival analysis. In *Survival and Event History Analysis*, ed. N. Keiding and P. K. Andersen, 266–271. Chichester, UK: Wiley.
- Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55. <https://doi.org/10.2307/2335942>.
- Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701. <https://doi.org/10.1037/h0037350>.
- Tsiatis, A. A. 2006. *Semiparametric Theory and Missing Data*. New York: Springer.
- Vittinghoff, E., D. V. Glidden, S. C. Shiboski, and C. E. McCulloch. 2012. *Regression Methods in Biostatistics: Linear, Logistic, Survival, and Repeated Measures Models*. 2nd ed. New York: Springer.
- Wooldridge, J. M. 2002. Inverse probability weighted M-estimators for sample selection, attrition, and stratification. *Portuguese Economic Journal* 1: 117–139. <https://doi.org/10.1007/s10258-002-0008-x>.
- . 2007. Inverse probability weighted estimation for general missing data problems. *Journal of Econometrics* 141: 1281–1301. <https://doi.org/10.1016/j.jeconom.2007.02.002>.
- . 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

## Also see

- [TE] [stteffects postestimation](#) — Postestimation tools for stteffects
- [TE] [stteffects intro](#) — Introduction to treatment effects for observational survival-time data
- [ST] [streg](#) — Parametric survival models
- [ST] [stset](#) — Declare data to be survival-time data
- [U] [20 Estimation and postestimation commands](#)

# Title

**tebalance** — Check balance after teffects or stteffects estimation

Description      Syntax      Remarks and examples      Methods and formulas  
References      Also see

## Description

The **tebalance** postestimation commands produce diagnostic statistics, test statistics, and diagnostic plots to assess whether a **teffects** or an **stteffects** command balanced the covariates over treatment levels.

## Syntax

```
tebalance subcommand ... [ , options ]
```

<i>subcommand</i>	Description
<b>summarize</b>	compare means and variances in raw and balanced data
<b>overid</b>	overidentification test
<b>density</b>	kernel density plots for raw and balanced data
<b>box</b>	box plots for each treatment level for balanced data

## Remarks and examples

This entry provides an overview of the commands in **tebalance**. We recommend that you read this entry before proceeding to [\[TE\] \*\*tebalance summarize\*\*](#), [\[TE\] \*\*tebalance overid\*\*](#), [\[TE\] \*\*tebalance density\*\*](#), or [\[TE\] \*\*tebalance box\*\*](#) for command-specific syntax and details.

A covariate is said to be balanced when its distribution does not vary over treatment levels.

Covariates are balanced in experimental data because treatment assignment is independent of the covariates because of the study design. In contrast, covariates must be balanced by weighting or matching in observational data because treatment assignment is related to the covariates that also affect the outcome of interest.

The estimators implemented in **teffects** and **stteffects** use a model or matching method to make the outcome conditionally independent of the treatment by conditioning on covariates. If this model or matching method is well specified, it should balance the covariates. Balance diagnostic techniques and tests check the specification of the conditioning method used by a **teffects** or an **stteffects** estimator; see [\[TE\] \*\*teffects intro advanced\*\*](#) for an introduction to **teffects**, and [\[TE\] \*\*stteffects intro\*\*](#) for an introduction to **stteffects**.

**tebalance** implements four methods to check for balance after **teffects** and **stteffects**. Which **tebalance** methods are available depends on the **teffects** estimation method, as summarized in the table below.



tebalance		Works after teffects					Works after stteffects	
method	Description	ipw	aipw	ipwra	nnmatch	psmatch	ipw	ipwra
summarize	standardized differences and variance ratios	x	x	x	x	x	x	x
overid	$\chi^2$ test for balance	x	x	x			x	x
density	diagnostic kernel-density plots	x	x	x	x	x	x	x
box	diagnostic box plots				x	x		

`tebalance overid` implements a formal test, while the other three methods are exploratory diagnostic techniques. There is no balance check after `teffects ra`, `stteffects ra`, or `stteffects wra`, because they use neither a treatment model nor a matching method.

Austin (2009, 2011) and Guo and Fraser (2015, sec. 5.52) provide introductions to covariate balance. Imai and Ratkovic (2014) derived a test for balance implemented in `tebalance overid`.

The remainder of this entry provides a quick introduction to using `tebalance` to check for balance after `teffects`. See [TE] [stteffects intro](#) for examples after `stteffects`.

## ► Example 1: Balance after estimators that use weighting

Inverse-probability-weighted (IPW) estimators use a model for the treatment to make the outcome conditionally independent of the treatment. If this model is well specified, it will also balance the covariates.

Using an extract from Cattaneo (2010), we use `teffects ipw` to estimate the effect of a mother's smoking behavior (`mbsmoke`) on the birthweight of her child (`bweight`), controlling for marital status (`mmarried`), the mother's age (`mage`), whether the mother had a prenatal doctor's visit in the baby's first trimester (`prenatal1`), and whether this baby is the mother's first child (`fbaby`).

```
. use https://www.stata-press.com/data/r17/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ipw (bweight) (mbsmoke mmarried mage prenatal1 fbaby)

Iteration 0:  EE criterion = 1.873e-22
Iteration 1:  EE criterion = 3.315e-26

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: logit
```

bweight	Robust				
	Coefficient	std. err.	z	P> z	[95% conf. interval]
ATE					
mbsmoke (Smoker vs Nonsmoker)	-236.1038	23.86187	-9.89	0.000	-282.8722 -189.3354
POmean					
mbsmoke Nonsmoker	3402.552	9.539555	356.68	0.000	3383.855 3421.249

Rubin (2008) recommends finding a model that balances the covariates before looking at results for the estimated treatment effect. Thus we do not interpret the above results, and we note that we could pay closer heed to Rubin’s recommendation by preceding the `teffects` command with `quietly` to suppress the output.

Imai and Ratkovic (2014) derived a test for balance by viewing the restrictions imposed by balance as overidentifying conditions. This test is implemented in `tebalance overid`, and we use it to test whether the above treatment model balanced the covariates.

```
. tebalance overid
Iteration 0: criterion = .02146858
Iteration 1: criterion = .02159149 (backed up)
Iteration 2: criterion = .02177783
Iteration 3: criterion = .02260102
Iteration 4: criterion = .02267956
Iteration 5: criterion = .02292367
Iteration 6: criterion = .02431655
Iteration 7: criterion = .02457028
Iteration 8: criterion = .02488569
Iteration 9: criterion = .02529483
Iteration 10: criterion = .0254588
Iteration 11: criterion = .02550245
Iteration 12: criterion = .02552864
Iteration 13: criterion = .02554462
Iteration 14: criterion = .02554512
Iteration 15: criterion = .02554514

Overidentification test for covariate balance
H0: Covariates are balanced

      chi2(5)      = 38.1464
      Prob > chi2  = 0.0000
```

We reject the null hypothesis that the treatment model balanced the covariates.

Let’s use `tebalance summarize` to see where the problem lies. To get an idea of the extent to which the covariates are unbalanced, we begin by summarizing the covariates by group in the raw data by specifying the `baseline` option.

```
. tebalance summarize, baseline
Covariate balance summary
```

	Raw	Weighted
Number of obs =	4,642	4,642.0
Treated obs =	864	2,315.3
Control obs =	3,778	2,326.7

	Means		Variances	
	Control	Treated	Control	Treated
mmarried	.7514558	.4733796	.1868194	.2495802
mage	26.81048	25.16667	31.87141	28.10429
prenatal1	.8268925	.6898148	.1431792	.2142183
fbaby	.4531498	.3715278	.2478707	.2337654

The output indicates that the covariates may not be balanced in the raw data. For example, the distribution of the mother’s age may differ over the treatment groups. We can investigate the differences further with standardized differences and variance ratios. A perfectly balanced covariate has a standardized difference of zero and variance ratio of one. There are no standard errors on these

statistics, so inference is informal. [Austin \(2009\)](#) provides a recent introduction to these diagnostics, although they have been used at least since [Rosenbaum and Rubin \(1985\)](#).

By omitting the `baseline` option, we obtain these diagnostic statistics for the raw data and the weighted data.

```
. tebalance summarize
```

```
Covariate balance summary
```

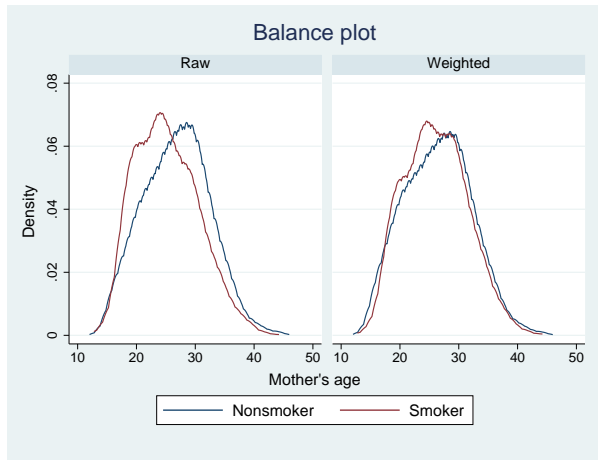
	Raw	Weighted
Number of obs =	4,642	4,642.0
Treated obs =	864	2,315.3
Control obs =	3,778	2,326.7

	Standardized differences		Variance ratio	
	Raw	Weighted	Raw	Weighted
mmarried	-.5953009	-.0105562	1.335944	1.009079
mage	-.300179	-.0672115	.8818025	.8536401
prenatal1	-.3242695	-.0156339	1.496155	1.023424
fbaby	-.1663271	.0257705	.9430944	1.005698

Reviewing the output, we see that for `mmarried`, `prenatal1`, and `fbaby`, our model improved the level of balance. The weighted standardized differences are all close to zero and the variance ratios are all close to one. However, output indicates that `mage` may not be balanced by our model. The weighted standardized difference is close to zero, but the weighted variance ratio still appears to be considerably less than one.

Now, let's use `tebalance density` to look at how the densities of `mage` for treated and control groups differ.

```
. tebalance density mage
```



The plots also indicate a lack of balance in `mage` between the treatment groups.

To try to achieve better balance, we specify a richer model with interactions between `mage` and the other covariates and look at the resulting standardized differences.

```
. quietly teffects ipw (bweight) (mbsmoke mmarried mage prenatal1 fbaby
> c.mage#(c.mage i.mmarried prenatal1))
```

```
. tebalance summarize
```

Covariate balance summary

	Raw	Weighted
Number of obs =	4,642	4,642.0
Treated obs =	864	2,329.1
Control obs =	3,778	2,312.9

	Standardized differences		Variance ratio	
	Raw	Weighted	Raw	Weighted
mmarried	-.5953009	.0053497	1.335944	.9953184
mage	-.300179	.0410889	.8818025	1.076571
prenatal1	-.3242695	.0009807	1.496155	.9985165
fbaby	-.1663271	-.0130638	.9430944	.9965406
mage#				
mage	-.3028275	.0477465	.8274389	1.109134
mmarried#				
mage				
Married	-.6329701	.0197209	1.157026	1.034108
prenatal1#				
mage				
Yes	-.4053969	.0182109	1.226363	1.032561

The standardized difference and variance ratio results for `mage` look closer to the expected values of zero and one, so we proceed to the formal test.

```
. tebalance overid
Iteration 0: criterion = .0602349
Iteration 1: criterion = .06172749 (backed up)
Iteration 2: criterion = .06428588 (backed up)
Iteration 3: criterion = .06489623 (backed up)
Iteration 4: criterion = .06527284 (backed up)
Iteration 5: criterion = .06643426
Iteration 6: criterion = .07134338
Iteration 7: criterion = .07638414
Iteration 8: criterion = .07673211
Iteration 9: criterion = .07681959
Iteration 10: criterion = .077044
Iteration 11: criterion = .07759547
Iteration 12: criterion = .07771973
Iteration 13: criterion = .0777271
Iteration 14: criterion = .07773395
Iteration 15: criterion = .07774839
Iteration 16: criterion = .07775314
Iteration 17: criterion = .07775324

Overidentification test for covariate balance
H0: Covariates are balanced
      chi2(8)      = 11.8612
      Prob > chi2  = 0.1575
```

We do not reject the null hypothesis that the specified treatment model balances the covariates. ◀

## ▶ Example 2: Balance after estimators that use matching

Instead of weighting, we might want to use a matching estimator. We can select `teffects nnmatch` or `teffects psmatch` for balance and estimation; in this example, we use `teffects nnmatch`.

```
. teffects nnmatch (bweight mmarried mage prenatal1 fbaby)
> (mbsmoke), bias(mage) ematch(mmarried prenatal1 fbaby)

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : nearest-neighbor matching      Matches: requested =      1
Outcome model  : matching                      min =      1
Distance metric: Mahalanobis                  max =      139
```

	bweight	AI robust		z	P> z	[95% conf. interval]
		Coefficient	std. err.			
ATE						
	mbsmoke (Smoker vs Nonsmoker)	-240.4589	28.43008	-8.46	0.000	-296.1808 -184.7369

Again we ignore the estimated effect and first check for balance. We begin by reviewing whether the summary statistics indicate good balance.

```
. tebalance summarize
(refitting the model using the generate() option)
```

Covariate balance summary

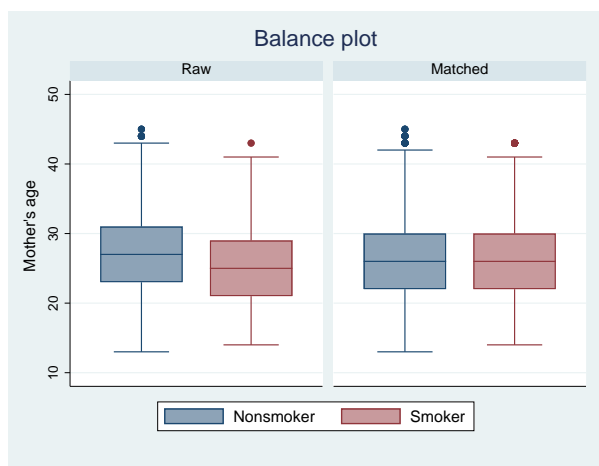
	Raw	Matched
Number of obs =	4,642	9,284
Treated obs =	864	4,642
Control obs =	3,778	4,642

	Standardized differences		Variance ratio	
	Raw	Matched	Raw	Matched
mmarried	-.5953009	-2.42e-16	1.335944	1
mage	-.300179	-.0040826	.8818025	.9815517
prenatall	-.3242695	-2.78e-16	1.496155	1
fbaby	-.1663271	2.24e-16	.9430944	1

We do not have standard errors on these statistics, so we cannot make any formal conclusions, but the summary statistics appear to indicate much better balance than the IPW results. `tebalance summarize` has to refit the model to recover the matched sample because the `generate()` option was not specified on the `teffects nnmatch` command. The reestimation does not affect the results, although the computation takes longer; see [example 3](#) for details.

Because it is a matching estimator, and not an IPW estimator, we cannot use `tebalance overid` after `teffects nnmatch`. The matching estimators, however, provide diagnostic box plots using `tebalance box` that are not available after the IPW estimators.

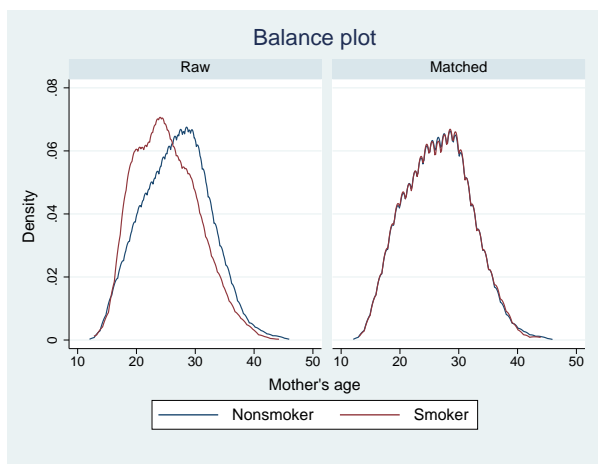
```
. tebalance box mage
(refitting the model using the generate() option)
```



The box plots of the matched data also indicate covariate balance.

Let's look at the kernel density plots using the matched data.

```
. tebalance density mage
(refitting the model using the generate() option)
```



The plots using the matched data appear to be balanced.

◀

## □ Technical note

`teffects` implements matching estimators, IPW estimators, regression-adjustment (RA) estimators, and estimators that combine IPW and RA. Matching estimators define a matched sample, and IPW estimators define a weighted sample, each of which can be used to compute covariate balance statistics. RA estimators do not define an adjusted sample that can be used to compute covariate balance statistics, and `tebalance` does not work after `teffects ra`. Only the IPW component of the estimators that combine RA and IPW defines a weighted sample that can be used to compute balance statistics. So, `tebalance` produces the same results after `teffects aipw` or `teffects ipwra` as it does after `teffects ipw`.

□

## ▷ Example 3: Faster results after a matching estimator

The `tebalance` commands run in [example 2](#) executed more slowly than necessary. `tebalance` issued the note

```
refitting the model using the generate() option
```

after the commands

```
. tebalance summarize
```

```
. tebalance box mage
```

and

```
. tebalance density mage
```

After `teffects nnmatch` or `teffects psmatch`, `tebalance` computes the balance statistics on the matched sample defined by the matching estimator. `teffects nnmatch` and `teffects psmatch` leave behind only variables that identify the matched sample when the `generate()` option is specified. Unless the `generate()` option is specified with `teffects nnmatch` or `teffects psmatch`, each `tebalance` command must rerun the estimation command to recover the matched sample.

### Typing

```
. teffects nnmatch (bweight mmarried mage fbaby prenatal1)
> (mbsmoke), bias(mage) ematch(mmarried fbaby prenatal1)
> generate(matchv)
```

would generate variables that identify the matched sample that the `tebalance` commands could use. See *Remarks and examples* in [TE] [tebalance box](#), [TE] [tebalance density](#), and [TE] [tebalance summarize](#) for examples using the option `generate()` on `teffects psmatch` to speed up the postestimation computations.

◀

## Methods and formulas

Methods and formulas are presented under the following headings:

- [Introduction](#)
- [Matched samples](#)
- [IPW samples](#)
- [Testing the propensity-score model specification](#)

### Introduction

For covariate  $z$ , we observe values  $\{z_1, z_2, \dots, z_N\}$ . Define a treatment indicator variable for  $J$  treatment levels as  $t_i \in \{1, 2, \dots, J\}$ , for  $i = 1, \dots, N$ , and frequency weights as  $\{w_1, w_2, \dots, w_N\}$ . The sample mean and variance of  $z$  for level  $t$  are

$$\hat{\mu}_z(t) = \frac{\sum_i^N I(t_i = t) w_i z_i}{N_t} \quad \text{and}$$

$$\hat{\sigma}_z^2(t) = \frac{\sum_i^N I(t_i = t) w_i \{z_i - \hat{\mu}_z(t)\}^2}{N_t - 1}$$

where  $N_t = \sum_i^N w_i I(t_i = t)$ , and

$$I(t_i = t) = \begin{cases} 1 & \text{if } t_i = t \\ 0 & \text{otherwise} \end{cases}$$

As shown in [Austin \(2011\)](#), the standardized differences for covariate  $z$  between level  $t$  and the control  $t_0$  are computed as

$$\delta_z(t) = \frac{\hat{\mu}_z(t) - \hat{\mu}_z(t_0)}{\sqrt{\frac{\hat{\sigma}_z^2(t) + \hat{\sigma}_z^2(t_0)}{2}}} \quad (1)$$

The variance ratio is  $\rho_z(t) = \{\hat{\sigma}_z^2(t)\} / \{\hat{\sigma}_z^2(t_0)\}$ .



## Matched samples

We now turn our attention to the matched samples for the potential-outcome mean (POM), average treatment effect (ATE), and average treatment effect on the treated (ATET) estimators. We estimate the covariate for the counter-factual treatment by taking the mean of the matched observations

$$\dot{z}_i = \frac{\sum_{j \in \Omega(i)} w_j z_j}{\sum_{j \in \Omega(i)} w_j}$$

where  $\Omega(i) = (k_1, k_2, \dots, k_{m_j})$  is the set of observation indices that are matched to observation  $i$  of the opposite treatment condition. The observed covariate and matched covariate pairs,  $(z_i, \dot{z}_i)$ ,  $i = 1, \dots, N$ , are used in the box plot (see [G-2] **graph box**) and the kernel density plot (see [R] **kdensity**). The ATET sample is limited to those observations from the conditional treatment,  $\tilde{t}$ , and their matched covariate means.

In *Methods and formulas* of [TE] **teffects nmatch**, we define  $K_m(i)$  as the number of times observation  $i$  is used in a match with observation  $j$  of the opposite treatment condition,  $i \in \Omega(j)$ , weighted by the total number of matches for observation  $j$ . Specifically,

$$K_m(i) = \sum_{j=1}^N I\{i \in \Omega(j)\} \frac{w_j}{\sum_{k \in \Omega(j)} w_k}$$

These weights are used in the estimation of the mean and variance for the matched dataset. For the POM and ATE models, the estimated mean and variance are computed as

$$\begin{aligned} \hat{\mu}_{\dot{z}}(t) &= \frac{\sum_i^N I(t_i = t) w_i z_i \{1 + K_m(i)\}}{M_t} \quad \text{and} \\ \hat{\sigma}_{\dot{z}}^2(t) &= \frac{\sum_i^N I(t_i = t) w_i \{1 + K_m(i)\} \{z_i - \hat{\mu}_{\dot{z}}(t)\}^2}{M_t - 1} \end{aligned}$$

where  $M_t = \sum_i^N I(t_i = t) w_i \{1 + K_m(i)\}$ .

The standardized differences between the control level and all other levels for the matched covariate distribution are computed as in (1), but  $\hat{\mu}_{\dot{z}}(t)$  is substituted for  $\hat{\mu}_z(t)$  and  $\hat{\sigma}_{\dot{z}}^2(t)$  for  $\hat{\sigma}_z^2(t)$ .

For the ATET model, there is no matched sample for the treatment levels other than the conditional treatment  $\tilde{t}$ . The covariate mean and variance for the conditional treatment are the same as that of the raw data,  $\mu_z(\tilde{t})$  and  $\sigma_z(\tilde{t})$ . However, the covariate mean and variance for the sample matched to the conditional treatment,  $t \neq \tilde{t}$ , are computed as

$$\begin{aligned} \tilde{\mu}_{\dot{z}}(t) &= \frac{\sum_i^N I(t_i = t) w_i z_i K_m(i)}{M_t} \quad \text{and} \\ \tilde{\sigma}_{\dot{z}}^2(t) &= \frac{\sum_i^N I(t_i = t) w_i K_m(i) \{z_i - \tilde{\mu}_{\dot{z}}(t)\}^2}{M_t - 1} \end{aligned}$$

where  $M_t = \sum_i^N I(t_i = t) w_i K_m(i)$ .

## IPW samples

Computation of the inverse-probability weights is discussed in *Methods and formulas* of [TE] **teffects aipw** and in *Methods and formulas* of [TE] **stteffects ipwra**. For the POM and ATE estimators, we defined the normalized IPW weights as  $\bar{d}_i(t) = N_t d_i(t) / \sum_i^N d_i(t)$ , where  $d_i(t) = I(t_i = t) / p(\mathbf{z}_i, t, \hat{\gamma})$  for treatment level  $t$  and individual  $i$ .

For the ATET estimator, we use the normalized weights  $\bar{f}_i = N f_i / \sum_i^N f_i$ , where  $f_i = p(\mathbf{z}_i, \tilde{t}, \hat{\gamma}) / p(\mathbf{z}_i, t_i, \hat{\gamma})$  are the treatment-adjusted inverse-probability weights, and  $\tilde{t}$  is the conditional treatment.

We will simplify notation by defining a single weight

$$\bar{w}_i(t) = \begin{cases} \bar{d}_i(t) & \text{if model is ATE or POM} \\ \bar{f}_i(t) & \text{if model is ATET} \end{cases}$$

The covariate mean and variance for treatment level  $t$  are

$$\begin{aligned} \tilde{\mu}_{\tilde{z}}(t) &= \frac{\sum_i^N I(t_i = t) w_i \bar{w}_i x_i}{M_t} & \text{and} \\ \tilde{\sigma}_{\tilde{z}}^2(t) &= \frac{I(t_i = t) w_i \bar{w}_i \{z_i - \tilde{\mu}_{\tilde{z}}(t)\}^2}{M_t - 1} \end{aligned}$$

where  $M_t = \sum_i^N I(t_i = t) w_i \bar{w}_i$ .

The kernel density is computed by `kdensity` for each covariate conditioned on each treatment level using the raw covariate with `iweights` equal to  $w_i \bar{w}_i$ .

## Testing the propensity-score model specification

We estimate the probability of treatment conditioned on a set of covariates with a propensity-score model. Imai and Ratkovic (2014) derive a test for whether the estimated propensity score balances the covariates. The score equations for parameters of the propensity-score model define an exactly identified generalized method of moments (GMM) estimator. Imai and Ratkovic (2014) use the conditions imposed by mean balance as overidentifying conditions. A standard GMM test for the validity of the overidentifying conditions is then a test for covariate balance. See [R] `gmm` for a discussion of this overidentifying test, which is known as Hansen's  $J$  test in the econometrics literature.

Here are the details about the score equations and the overidentifying balance conditions. Recall from *Methods and formulas* of [TE] **teffects aipw** and *Methods and formulas* of [TE] **stteffects ipwra**, we have the first-order condition of the treatment model

$$\frac{1}{N} \sum_{i=1}^N s_{\text{tm},i}(\mathbf{z}_i, \hat{\gamma}) = 0$$

For a two-level treatment-effects model with conditional treatment  $\tilde{t}$  and control  $t_0$ , the score is

$$s_{\text{tm},i}(\mathbf{z}_i, \gamma) = \frac{I(t_i = \tilde{t})}{p(\mathbf{z}_i, \tilde{t}, \gamma)} \frac{\partial p(\mathbf{z}_i, t, \gamma)}{\partial \gamma'} - \left\{ \frac{I(t_i = t_0)}{1 - p(\mathbf{z}_i, \tilde{t}, \gamma)} \right\} \frac{\partial p(\mathbf{z}_i, \tilde{t}, \gamma)}{\partial \gamma'} \Bigg|_{\gamma = \hat{\gamma}}$$

The score reduces to

$$\mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \tilde{\gamma}) = \left[ \frac{I(t_i = \tilde{t}) - p(\mathbf{z}_i, \tilde{t}, \gamma)}{p(\mathbf{z}_i, \tilde{t}, \gamma) \{1 - p(\mathbf{z}_i, \tilde{t}, \gamma)\}} \right] \frac{\partial p(\mathbf{z}_i, \tilde{t}, \gamma)}{\partial \gamma'} \Bigg|_{\gamma = \tilde{\gamma}}$$

The corresponding covariate balancing moment conditions are

$$\mathbf{w}_{\text{tm},i}(\mathbf{z}_i, \gamma) = \left[ \frac{I(t_i = \tilde{t}) - p(\mathbf{z}_i, \tilde{t}, \gamma)}{p(\mathbf{z}_i, \tilde{t}, \gamma) \{1 - p(\mathbf{z}_i, \tilde{t}, \gamma)\}} \right] \mathbf{z}_i$$

for the POM and ATE models. For the ATET model with conditional treatment  $\tilde{t}$ , we multiply by  $p(\mathbf{z}_i, \tilde{t}, \gamma)$  and scale by  $N/N_{\tilde{t}}$ :

$$\mathbf{w}_{\text{tm},i}(\mathbf{z}_i, \gamma) = \frac{N}{N_{\tilde{t}}} \left\{ \frac{I(t_i = \tilde{t}) - p(\mathbf{z}_i, \tilde{t}, \gamma)}{1 - p(\mathbf{z}_i, \tilde{t}, \gamma)} \right\} \mathbf{z}_i$$

We stack the moment conditions

$$\begin{aligned} \mathbf{g}_{\text{tm}}(\mathbf{Z}, \gamma) &= \frac{1}{N} \sum_{i=1}^N \begin{Bmatrix} \mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \gamma) \\ \mathbf{w}_{\text{tm},i}(\mathbf{z}_i, \gamma) \end{Bmatrix} \\ &= \frac{1}{N} \sum_{i=1}^N \mathbf{g}_{\text{tm},i}(\mathbf{z}_i, \gamma) \end{aligned}$$

The overidentified GMM estimator is then

$$\tilde{\gamma} = \operatorname{argmin}_{\gamma} N \mathbf{g}_{\text{tm}}(\mathbf{Z}, \gamma)' \mathbf{W}_{\text{tm}}(\mathbf{Z}, \gamma)^{-1} \mathbf{g}_{\text{tm}}(\mathbf{Z}, \gamma) \quad (2)$$

where

$$\mathbf{W}_{\text{tm}}(\mathbf{Z}, \gamma) = \frac{1}{N} \sum_{i=1}^N E_T \{ \mathbf{g}_{\text{tm},i}(\mathbf{z}, \gamma) \mathbf{g}_{\text{tm},i}(\mathbf{z}, \gamma)' \}$$

and the expectation is taken with respect to treatment distribution. The weight matrix  $\mathbf{W}_{\text{tm}}(\mathbf{Z}, \gamma)$  is computed explicitly (Imai and Ratkovic 2014), and (2), written as a maximization problem, is solved using `m1`.

Finally, Hansen's  $J$  statistic is evaluated at its minimum,

$$J = N \mathbf{g}_{\text{tm}}(\mathbf{Z}, \tilde{\gamma})' \mathbf{W}_{\text{tm}}(\mathbf{Z}, \tilde{\gamma})^{-1} \mathbf{g}_{\text{tm}}(\mathbf{Z}, \tilde{\gamma})$$

and is asymptotically distributed  $\chi^2$  with degrees of freedom  $d$ ,

$$d = \operatorname{rank} \{ \mathbf{W}_{\text{tm}}(\mathbf{Z}, \tilde{\gamma}) \} - \operatorname{rank} \left[ \frac{1}{N} \sum_{i=1}^N E_T \{ \mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \tilde{\gamma}) \mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \tilde{\gamma})' \} \right]$$

## References

- Austin, P. C. 2009. Balance diagnostics for comparing the distribution of baseline covariates between treatment groups in propensity-score matched samples. *Statistics in Medicine* 28: 3083–3107. <https://doi.org/10.1002/sim.3697>.
- . 2011. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate Behavioral Research* 46: 399–424. <https://doi.org/10.1080/00273171.2011.568786>.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Guo, S., and M. W. Fraser. 2015. *Propensity Score Analysis: Statistical Methods and Applications*. 2nd ed. Thousand Oaks, CA: SAGE.
- Imai, K., and M. Ratkovic. 2014. Covariate balancing propensity score. *Journal of the Royal Statistical Society, Series B* 76: 243–263. <https://doi.org/10.1111/rssb.12027>.
- Jann, B. 2021. [Relative distribution analysis in Stata](#). *Stata Journal* 21: 885–951.
- Rosenbaum, P. R., and D. B. Rubin. 1985. Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. *American Statistician* 39: 33–38. <https://doi.org/10.2307/2683903>.
- Rubin, D. B. 2008. For objective causal inference, design trumps analysis. *Annals of Applied Statistics* 2: 808–840. <https://doi.org/10.1214/08-AOAS187>.

## Also see

- [TE] [stteffects intro](#) — Introduction to treatment effects for observational survival-time data
- [TE] [tebalance box](#) — Covariate balance box
- [TE] [tebalance density](#) — Covariate balance density
- [TE] [tebalance overid](#) — Test for covariate balance
- [TE] [tebalance summarize](#) — Covariate-balance summary statistics
- [TE] [teffects](#) — Treatment-effects estimation for observational data

# Title

**tebalance box** — Covariate balance box

[Description](#)  
[Options](#)

[Quick start](#)  
[Remarks and examples](#)

[Menu](#)  
[Reference](#)

[Syntax](#)  
[Also see](#)

## Description

`tebalance box` produces box plots that are used to check for balance in matched samples after `teffects nnmatch` and `teffects psmatch`.

## Quick start

Box plot of the propensity score from the last `teffects psmatch` command

```
tebalance box
```

Box plot of values of `x1` in the treatment and control groups from raw data and the matched sample from the last `teffects nnmatch` or `teffects psmatch` command

```
tebalance box x1
```

## Menu

Statistics > Treatment effects > Balance > Graphs

## Syntax

Box plots for the propensity score

```
tebalance box [ , options ]
```

Box plots for a covariate

```
tebalance box varname [ , options ]
```

*options*

Description

Main

*boxlook\_options*

**graph box** options controlling how the box looks

*legending\_options*

**graph box** options controlling how the variables are labeled

*axis\_options*

**graph box** options controlling how numerical *y* axis is labeled

*title\_and\_other\_options*

**graph box** options controlling titles, added text, aspect ratio, etc.

*by\_options*

suboptions inside `by()` controlling plots by raw and matched samples

## Options

Main

*boxlook\_options* are any of the options documented in *boxlook\_options* in [G-2] **graph box**.

*legending\_options* are any of the options documented in *legending\_options* in [G-2] **graph box**.

*axis\_options* are any of the options documented in *axis\_options* in [G-2] **graph box**.

*title\_and\_other\_options* are any of the options, except `by()`, documented in *title\_and\_other\_options* in [G-2] **graph box**. `tebalance box` uses `by()` to differentiate between raw and matched samples, and some *twoway\_options* will be repeated for `by graph` and might be better specified as `byopts()`.

*by\_options* are any of the *byopts* documented in [G-3] *by\_option*. `byopts()` generally affects the entire graph, and some *by\_options* may be better specified as *twoway\_options*; see [G-3] **twoway\_options**.

## Remarks and examples

When the distribution of a covariate does not vary over the treatment levels, the covariate is said to be balanced. `tebalance box` produces box plots of a covariate over treatment levels for the raw data and for the matched sample produced by `teffects`. If the matched-sample box plots are the same over the treatment levels, the covariate is balanced in the matched sample.

After `teffects nnmatch` and `teffects psmatch`,

```
. tebalance box varname [ , options ]
```

produces box plots to check whether *varname* is balanced.

After `teffects psmatch`,

```
. tebalance box [, options]
```

produces box plots to check whether the propensity score estimated by `teffects` is balanced.

We recommend that you read [TE] [tebalance](#) before proceeding; it provides an introduction to covariate balance and an overview of the implemented methods.

### ▷ Example 1: Checking covariate balance after `psmatch`

Using an extract from the data used by [Cattaneo \(2010\)](#), we use `teffects psmatch` to estimate the effect of a mother's smoking behavior (`mbsmoke`) on the birthweight of her child (`bweight`), controlling for marital status (`mmarried`), the mother's age (`mage`), whether the mother had a prenatal doctor's visit in the baby's first trimester (`prenatal1`), and whether this baby is the mother's first child (`fbaby`).

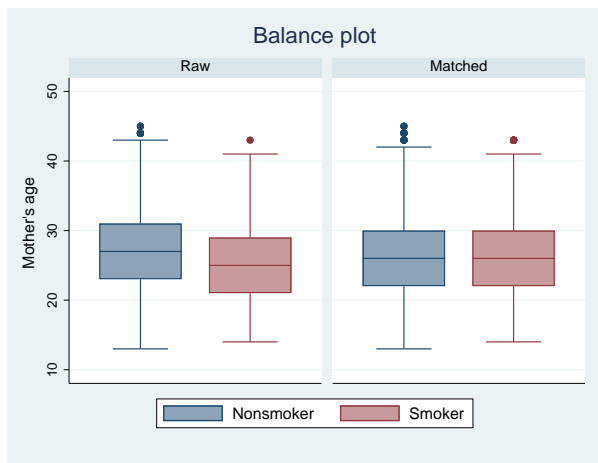
```
. use https://www.stata-press.com/data/r17/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects psmatch (bweight) (mbsmoke mmarried mage prenatal1 fbaby),
> generate(matchv)
```

```
Treatment-effects estimation      Number of obs      =      4,642
Estimator      : propensity-score matching      Matches: requested =      1
Outcome model  : matching                      min =      1
Treatment model: logit                        max =      139
```

bweight		AI robust		z	P> z	[95% conf. interval]
	Coefficient	std. err.				
ATE						
mbsmoke (Smoker vs Nonsmoker)	-235.1714	27.74409	-8.48	0.000	-289.5488	-180.794

We specified the option `generate(matchv)` to speed up the postestimation command that produces density plots, as discussed in [example 3](#) under *Remarks and examples* of [TE] [tebalance](#). We do not interpret the estimated effect produced by this preliminary model but rather check the specification. Now we look at the box plots.

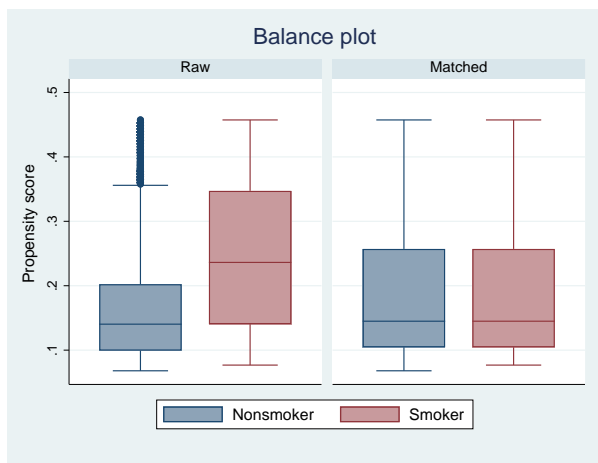
```
. tebalance box mage
```



The box plots for the matched sample are very similar. The medians, the 25th percentiles, and the 75th percentiles appear to be the same, although there may be some differences in the tails, the upper adjacent values, the lower adjacent values, and the outliers. Matching on the estimated propensity score appears to have balanced `mage`, except for the tails.

To get an idea of whether `teffects psmatch` balanced all the covariates, we look at the box plots for the estimated propensity score.

```
. tebalance box
```



The box plots indicate that `teffects psmatch` balanced the estimated propensity scores.



## Reference

Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.

## Also see

[TE] [tebalance](#) — Check balance after teffects or stteffects estimation

[TE] [teffects nnmatch](#) — Nearest-neighbor matching

[TE] [teffects psmatch](#) — Propensity-score matching

[TE] [teoverlap](#) — Overlap plots

# Title

**tebalance density** — Covariate balance density

[Description](#)  
[Options](#)  
[Also see](#)

[Quick start](#)  
[Remarks and examples](#)

[Menu](#)  
[Stored results](#)

[Syntax](#)  
[Reference](#)

## Description

`tebalance density` produces kernel density plots that are used to check for covariate balance after estimation by a `teffects` inverse-probability-weighted estimator, a `teffects` matching estimator, or an `stteffects` inverse-probability-weighted estimator.

## Quick start

Kernel density plot of the propensity score after `teffects psmatch`

```
tebalance density
```

Kernel density plot of `x1` after a `teffects` command or an `stteffects` command

```
tebalance density x1
```

As above, but rescale the kernel bandwidth by a factor of 2

```
tebalance density x1, bwidth(*2)
```

## Menu

Statistics > Treatment effects > Balance > Graphs

## Syntax

Density plots for the propensity score

```
tebalance density [, options]
```

Density plots for a covariate

```
tebalance density varname [, options]
```

<i>options</i>	Description
----------------	-------------

Main

<code>kernel(<i>kernel</i>)</code>	specify the kernel function; default is <code>kernel(epanechnikov)</code>
<code>bwidth(*#)</code>	rescale default bandwidth
<code>line#opts(<i>line_options</i>)</code>	twoway line options for density line number #
<code>twoway_options</code>	any options other than <code>by()</code> documented in [G-3] <i>twoway_options</i>
<code>byopts(<i>byopts</i>)</code>	how subgraphs are combined, labeled, etc.

collect is allowed; see [U] 11.1.10 Prefix commands.

<i>kernel</i>	Description
---------------	-------------

<code>triangle</code>	triangle kernel function; the default
<code>epanechnikov</code>	Epanechnikov kernel function
<code>epan2</code>	alternative Epanechnikov kernel function
<code>biweight</code>	biweight kernel function
<code>cosine</code>	cosine trace kernel function
<code>gaussian</code>	Gaussian kernel function
<code>parzen</code>	Parzen kernel function
<code>rectangle</code>	rectangle kernel function

## Options

Main

`kernel(kernel)` specifies the kernel function for use in calculating the kernel density estimates. The default kernel is the `kernel(epanechnikov)`.

`bwidth(*#)` specifies the factor by which the default bandwidths are to be rescaled. A bandwidth is the half-width of the kernel, the width of the density window around each point. Each kernel density plot has its own bandwidth, and by default, each kernel density plot uses its own optimal bandwidth; see [R] [kdensity](#). `bwidth()` rescales each plot's optimal bandwidth by the specified amount.

`line#opts(line_options)` specifies the line pattern, width, color, and overall style of density line number #. The line numbers are in the same order as the treatment levels specified in `e(tlevels)`.

`twoway_options` are any of the options documented in [G-3] *twoway\_options*, excluding `by()`. These include options for titling the graph (see [G-3] *title\_options*) and for saving the graph to disk (see [G-3] *saving\_option*). `tebalance density` uses `by()` to differentiate between raw and weighted or matched samples, and some *twoway\_options* will be repeated for `by` graph and might be better specified as `byopts()`.

`byopts` (*by\_option*) is as documented in [G-3] [by\\_option](#). `byopts()` affects how the subgraphs are combined, labeled, etc. `byopts()` generally affects the entire graph, and some *by\_option* may be better specified as *twoway\_options*; see [G-3] [twoway\\_options](#).

## Remarks and examples

When the distribution of a covariate does not vary over the treatment levels, the covariate is said to be balanced. `tebalance density` produces kernel density plots of a covariate over treatment levels for the raw data and the weighted or matched sample produced by `teffects` or `stteffects`. If the weighted-sample or matched-sample kernel density plots of the covariate are the same over the treatment levels, the covariate is balanced in the weighted or matched sample.

After `telasso` and all `teffects` and `stteffects` commands except `teffects ra`, `stteffects ra`, and `stteffects wra`,

```
. tebalance density varname [, options]
```

produces kernel density plots to check whether *varname* is balanced.

After `teffects psmatch`,

```
. tebalance density [, options]
```

produces kernel density plots to check whether the propensity score estimated by `teffects psmatch` is balanced. Our discussion of the use of `tebalance density` and interpretation of its results for a covariate below also apply to a propensity score.

We recommend that you read [TE] [tebalance](#) before proceeding; it provides an introduction to covariate balance and an overview of the implemented methods. See [TE] [stteffects intro](#) for a discussion of survival-time features.

### ► Example 1: Checking covariate balance after `psmatch`

Using an extract from the data used by [Cattaneo \(2010\)](#), we use `teffects psmatch` to estimate the effect of a mother's smoking behavior (`mbsmoke`) on the birthweight of her child (`bweight`), controlling for marital status (`mmarried`), the mother's age (`mage`), whether the mother had a prenatal doctor's visit in the baby's first trimester (`prenatal1`), and whether this baby is the mother's first child (`fbaby`).

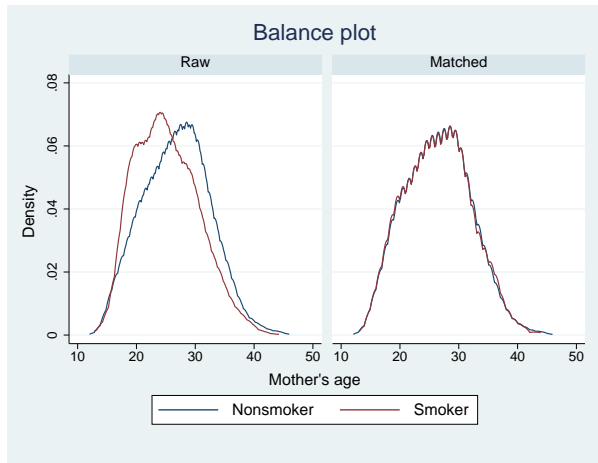
```
. use https://www.stata-press.com/data/r17/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects psmatch (bweight) (mbsmoke mmarried mage prenatal1 fbaby),
> generate(matchv)
```

```
Treatment-effects estimation      Number of obs      =      4,642
Estimator      : propensity-score matching      Matches: requested =      1
Outcome model  : matching                      min =      1
Treatment model: logit                        max =     139
```

		AI robust				
	bweight	Coefficient	std. err.	z	P> z	[95% conf. interval]
ATE						
	mbsmoke					
	(Smoker					
	vs					
	Non smoker)	-235.1714	27.74409	-8.48	0.000	-289.5488 -180.794

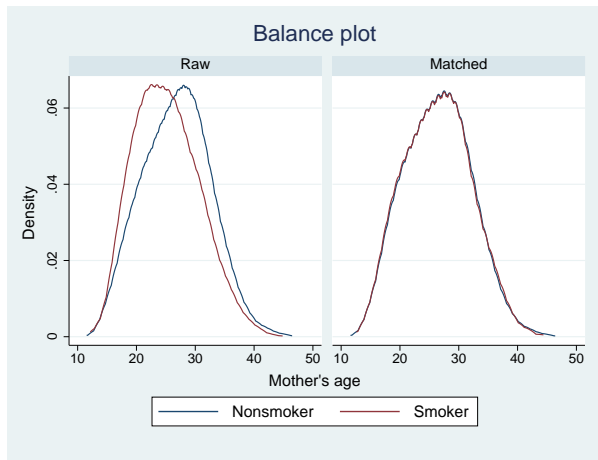
We specified the option `generate(matchhv)` to speed up the postestimation command that produces density plots, as discussed in [example 3](#) under *Remarks and examples* in [TE] `tebalance` entry. We do not interpret the estimated effect produced by this preliminary model but rather check the specification. We begin by looking at the default density plots.

```
. tebalance density mage
```



The density plots for the matched sample are nearly indistinguishable, implying that matching on the estimated propensity score balanced the covariates. The density plots are too jagged for presentation, so we oversmooth them by scaling up the bandwidth used for each plot.

```
. tebalance density mage, bwidth(*1.5)
```



Option `bwidth()` rescales the default optimal bandwidths by the specified scale factor. Each of the four density plots has its own sample size and optimal bandwidth. Rescaling each of the four bandwidths by 1.5 produces smoother plots.

## Stored results

After `teffects` or `stteffects` fits a binary treatment, `tebalance density` stores the following in `r()`:

### Scalars

<code>r(bwc_adj)</code>	bandwidth for control in weighted or matched-adjusted sample
<code>r(Nc_adj)</code>	observations on control in weighted or matched-adjusted sample
<code>r(bwt_adj)</code>	bandwidth for treated in weighted or matched-adjusted sample
<code>r(Nt_adj)</code>	observations on treated in weighted or matched-adjusted sample
<code>r(bwc_raw)</code>	bandwidth for control in raw sample
<code>r(Nc_raw)</code>	observations on control in raw sample
<code>r(bwt_raw)</code>	bandwidth for treated in raw sample
<code>r(Nt_raw)</code>	observations on treated in raw sample

### Macros

<code>r(kernel)</code>	name of kernel
------------------------	----------------

After `teffects` or `stteffects` fits a multivalued treatment, `tebalance density` stores the following in `r()`:

### Scalars

<code>r(bw#_adj)</code>	bandwidth for treatment level # in weighted or matched-adjusted sample
<code>r(N#_adj)</code>	observations on treatment level # in weighted or matched-adjusted sample
<code>r(bw#_raw)</code>	bandwidth for treatment level # in raw sample
<code>r(N#_raw)</code>	observations on treatment level # in raw sample

### Macros

<code>r(kernel)</code>	name of kernel
------------------------	----------------

## Reference

Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.

## Also see

- [TE] [stteffects intro](#) — Introduction to treatment effects for observational survival-time data
- [TE] [tebalance](#) — Check balance after `teffects` or `stteffects` estimation
- [TE] [teffects aipw](#) — Augmented inverse-probability weighting
- [TE] [teffects ipw](#) — Inverse-probability weighting
- [TE] [teffects ipwra](#) — Inverse-probability-weighted regression adjustment
- [TE] [teffects nnmatch](#) — Nearest-neighbor matching
- [TE] [teffects psmatch](#) — Propensity-score matching
- [TE] [teoverlap](#) — Overlap plots

# Title

**tebalance overid** — Test for covariate balance

[Description](#)  
[Options](#)  
[Also see](#)

[Quick start](#)  
[Remarks and examples](#)

[Menu](#)  
[Stored results](#)

[Syntax](#)  
[References](#)

## Description

`tebalance overid` performs a test for covariate balance after estimation by a `teffects` inverse-probability-weighted (IPW) estimator or an `stteffects` IPW estimator.

## Quick start

Test for covariate balance after a `teffects` or an `stteffects` IPW estimator

```
tebalance overid
```

As above, but test for balance only in base covariates and exclude interaction terms

```
tebalance overid, bconly
```

## Menu

Statistics > Treatment effects > Balance > Overidentification test

## Syntax

```
tebalance overid [ , bconly nolog iterate(#) ]
```

`collect` is allowed; see [U] 11.1.10 **Prefix commands**.

## Options

### Main

**bconly** specifies that only the base covariates be included in the test for balance. By default, the powers and interactions specified by factor-variable notation in the **teffects** or **stteffects** model are also included in the test for balance.

**nolog** suppresses the display of the optimization search log.

**iterate**(#) sets the maximum number of iterations to # in the generalized method of moments estimator used to compute the test statistic.

## Remarks and examples

When the distribution of a covariate is the same for all treatment levels, the covariate is said to be balanced. **tebalance overid** performs a test to see whether the covariates are balanced after **teffects** or **stteffects**. **tebalance overid** can be executed after **teffects ipw**, **teffects aipw**, **teffects ipwra**, **stteffects ipw**, or **stteffects ipwra**, which use the inverse-probability weights predicted by a treatment model to account for how treatment assignment depends on observed covariates. If the treatment model is well specified, IPW functions of the covariates from the model are balanced.

We recommend that you read [TE] **tebalance** before proceeding; it provides an introduction to covariate balance and an overview of the implemented methods. See [TE] **stteffects intro** for survival-time discussion and examples.

### ► Example 1: Base covariates and interactions

This example illustrates the interpretation of the **bconly** option, which excludes powers and interactions when factor variables are included in the propensity-score model.

We frequently use factor variables to include powers of, and interactions between, base covariates in our specification of the propensity-score model. In [example 1](#) under *Remarks and examples* in [TE] **tebalance**, we rejected the null hypothesis of balance in a model using only base covariates but not in the richer model that included power and interaction terms. By default, **tebalance overid** tests whether the model balances the base covariates and the power-and-interaction covariates. When option **bconly** is specified, **tebalance overid** tests whether the model balances the base covariates only.

Using an extract from the data used by [Cattaneo \(2010\)](#), we use **teffects ipw** to estimate the effect of a mother's smoking behavior (**mbsmoke**) on the birthweight of her child (**bweight**), controlling for marital status (**mmarried**), the mother's age (**mage**), whether the mother had a prenatal doctor's visit in the baby's first trimester (**prenatal1**), and whether this baby is the mother's first child (**fbaby**). In addition to the base covariates, we include the square of **mage**, an interaction between **mage** and **mmarried**, and an interaction between **mage** and **prenatal1** in the model for the propensity score.



```

. use https://www.stata-press.com/data/r17/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ipw (bweight) (mbsmoke mmarried mage prenatal1 fbaby
> c.mage#c.mage i.mmarried prenatal1), aequations
Iteration 0:  EE criterion = 9.365e-20
Iteration 1:  EE criterion = 2.612e-26
Treatment-effects estimation          Number of obs    =    4,642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: logit

```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
<b>ATE</b>						
mbsmoke (Smoker vs Nonsmoker)	-239.6875	26.43427	-9.07	0.000	-291.4977	-187.8773
<b>POmean</b>						
mbsmoke Nonsmoker	3403.638	9.56792	355.73	0.000	3384.885	3422.39
<b>TME1</b>						
mmarried	.8522468	.462536	1.84	0.065	-.0543072	1.758801
mage	.1742823	.0651039	2.68	0.007	.0466811	.3018836
prenatal1	.4018114	.4341762	0.93	0.355	-.4491584	1.252781
fbaby	-.4824413	.0868982	-5.55	0.000	-.6527587	-.3121239
c.mage# c.mage	-.002515	.0012585	-2.00	0.046	-.0049817	-.0000483
mmarried# c.mage Married	-.0787984	.0175508	-4.49	0.000	-.1131973	-.0443996
prenatal1# c.mage Yes	-.0286228	.0176391	-1.62	0.105	-.0631948	.0059492
_cons	-2.928851	.8409119	-3.48	0.000	-4.577008	-1.280694

We specified option `aequations` to see the parameter estimates for the coefficients in the propensity-score model. There are eight coefficients, five on the base covariates (`mmarried`, `mage`, `fbaby`, `prenatal1`, and `_cons`) and three on the power-and-interaction covariates (`c.mage#c.mage`, `c.mage#1.mmarried`, and `c.mage#1.prenatal1`). Below we test whether the model balances all eight covariates.

```

. tebalance overid

Iteration 0:  criterion =  .0602349
Iteration 1:  criterion =  .06172749 (backed up)
Iteration 2:  criterion =  .06428588 (backed up)
Iteration 3:  criterion =  .06489623 (backed up)
Iteration 4:  criterion =  .06527284 (backed up)
Iteration 5:  criterion =  .06643426
Iteration 6:  criterion =  .07134338
Iteration 7:  criterion =  .07638414
Iteration 8:  criterion =  .07673211
Iteration 9:  criterion =  .07681959
Iteration 10: criterion =  .077044
Iteration 11: criterion =  .07759547
Iteration 12: criterion =  .07771973
Iteration 13: criterion =  .0777271
Iteration 14: criterion =  .07773395
Iteration 15: criterion =  .07774839
Iteration 16: criterion =  .07775314
Iteration 17: criterion =  .07775324

Overidentification test for covariate balance
H0: Covariates are balanced

      chi2(8)      =  11.8612
      Prob > chi2  =  0.1575

```

We cannot reject the null hypothesis that the IPW model balanced all eight covariates.

Below we specify option `bconly` to test whether the IPW model balanced the five base covariates only.

```

. tebalance overid, bconly

Iteration 0:  criterion =  .1079977
Iteration 1:  criterion =  .10800825 (backed up)
Iteration 2:  criterion =  .10844177 (backed up)
Iteration 3:  criterion =  .10851228 (backed up)
Iteration 4:  criterion =  .10860856 (backed up)
Iteration 5:  criterion =  .10907494
Iteration 6:  criterion =  .11009792
Iteration 7:  criterion =  .11164102
Iteration 8:  criterion =  .11260711
Iteration 9:  criterion =  .11286467
Iteration 10: criterion =  .11331528
Iteration 11: criterion =  .11333981
Iteration 12: criterion =  .1133561
Iteration 13: criterion =  .11335696
Iteration 14: criterion =  .11335696

Overidentification test for covariate balance
H0: Covariates are balanced

      chi2(5)      =  7.82169
      Prob > chi2  =  0.1663

```

We cannot reject the null hypothesis that the IPW model balanced the five base covariates.

Each test has a justification.

In a model-based approach, the Imai and Ratkovic (2014) test checks whether the propensity score is correctly specified. We include all eight covariates because they must all be balanced, if the propensity-score model is correctly specified.

A conditional-independence approach can be used to justify only including the base covariates in the test. In this approach, the propensity-score model need only balance the base covariates. Powers and interactions of the base covariates are included to get a propensity-score model that balances the base covariates, but balance of these higher-order terms is more than what needs to be checked.

In large samples, both tests should have nominal coverage under the null hypothesis that the propensity-score model is correctly specified. Under the alternative that the propensity-score model is misspecified, including all the covariates should yield a test with higher power.

The test that includes all the covariates is the default.



## Stored results

`tebalance overid` stores the following in `r()`:

Scalars

<code>r(p)</code>	<i>p</i> -value
<code>r(df)</code>	overidentifying constraints, test degrees of freedom
<code>r(chi2)</code>	$\chi^2$ statistic

## References

- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Imai, K., and M. Ratkovic. 2014. Covariate balancing propensity score. *Journal of the Royal Statistical Society, Series B* 76: 243–263. <https://doi.org/10.1111/rssb.12027>.

## Also see

- [TE] [stteffects intro](#) — Introduction to treatment effects for observational survival-time data
- [TE] [tebalance](#) — Check balance after `teffects` or `stteffects` estimation
- [TE] [teffects aipw](#) — Augmented inverse-probability weighting
- [TE] [teffects ipw](#) — Inverse-probability weighting
- [TE] [teffects ipwra](#) — Inverse-probability-weighted regression adjustment
- [TE] [teoverlap](#) — Overlap plots

# Title

**tebalance summarize** — Covariate-balance summary statistics

[Description](#)  
[Option](#)  
[Also see](#)

[Quick start](#)  
[Remarks and examples](#)

[Menu](#)  
[Stored results](#)

[Syntax](#)  
[Reference](#)

## Description

`tebalance summarize` reports diagnostic statistics that are used to check for covariate balance over treatment groups after estimation by a `teffects` inverse-probability-weighted (IPW) estimator, a `teffects` matching estimator, or an `stteffects` IPW estimator.

## Quick start

Raw and weighted standardized differences and variance ratios of all covariates from the most recently estimated `teffects` model or `stteffects` model

```
tebalance summarize
```

As above, but report statistics only for covariates `x1` and `x2`

```
tebalance summarize x1 x2
```

Baseline means and variances for treated and control groups

```
tebalance summarize, baseline
```

## Menu

[Statistics](#) > [Treatment effects](#) > [Balance](#) > [Summaries](#)

## Syntax

```
tebalance summarize [varlist] [, baseline]
```

*varlist* may contain factor variables; see [U] 11.4.3 Factor variables.

collect is allowed; see [U] 11.1.10 Prefix commands.

## Option

Main

*baseline* specifies that `tebalance summarize` report means and variances by treatment level.

## Remarks and examples

When the distribution of a covariate is the same for all treatment levels, the covariate is said to be balanced. `tebalance summarize` reports diagnostic statistics to check for covariate balance after `teffects` or `stteffects`. `tebalance summarize` can be executed after all `teffects` estimators with the exception of `teffects ra` and executed after `stteffects ipw` and `stteffects ipwra`.

We recommend that you read [TE] [tebalance](#) before proceeding; it provides an introduction to covariate balance and an overview of the implemented methods. See [TE] [stteffects intro](#) for survival-time discussion and examples.

### ► Example 1: Checking covariate balance after `psmatch`

Using an extract from the data used by Cattaneo (2010), we use `teffects psmatch` to estimate the effect of a mother's smoking behavior (`mbsmoke`) on the birthweight of her child (`bweight`), controlling for marital status (`mmarried`), the mother's age (`mage`), whether the mother had a prenatal doctor's visit in the baby's first trimester (`prenatal1`), and whether this baby is the mother's first child (`fbaby`).

```
. use https://www.stata-press.com/data/r17/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects psmatch (bweight) (mbsmoke mmarried mage prenatal1 fbaby),
> generate(matchv)

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : propensity-score matching      Matches: requested =      1
Outcome model  : matching                      min =      1
Treatment model: logit                        max =      139
```

		AI robust				
	bweight	Coefficient	std. err.	z	P> z	[95% conf. interval]
ATE						
	mbsmoke					
	(Smoker					
	vs					
	Non smoker)	-235.1714	27.74409	-8.48	0.000	-289.5488 -180.794

We specified the option `generate(matchv)` to speed up the postestimation commands that compute balance statistics, as discussed in [example 3](#) under *Remarks and examples* in [\[TE\] tebalance](#). We do not interpret the estimated effect produced by this preliminary model but rather check the specification.

We begin by looking at the standardized differences and variance ratios for the raw data and the matched sample.

```
. tebalance summarize
Covariate balance summary
```

	Raw	Matched
Number of obs =	4,642	9,284
Treated obs =	864	4,642
Control obs =	3,778	4,642

	Standardized differences		Variance ratio	
	Raw	Matched	Raw	Matched
mmarried	-.5953009	.0014107	1.335944	.9987659
mage	-.300179	-.0120277	.8818025	.9952916
prenatal1	-.3242695	.0333609	1.496155	.9491524
fbaby	-.1663271	-.0117326	.9430944	.9969095

The matched sample results indicate that matching on the estimated propensity score balanced the covariates. The standardized differences are all close to zero, and the variance ratios are all close to one. This inference is informal because we do not have standard errors for these statistics.

We may also wish to see the baseline summary statistics.

```
. tebalance summarize, baseline
Covariate balance summary
```

	Raw	Matched
Number of obs =	4,642	9,284
Treated obs =	864	4,642
Control obs =	3,778	4,642

	Means		Variances	
	Control	Treated	Control	Treated
mmarried	.7514558	.4733796	.1868194	.2495802
mage	26.81048	25.16667	31.87141	28.10429
prenatal1	.8268925	.6898148	.1431792	.2142183
fbaby	.4531498	.3715278	.2478707	.2337654

While we rely on the standardized differences for conclusions about balance in the unmatched sample from this output, the baseline means and variances give us some idea of the scale of the differences.

## ► Example 2: Multivalued treatments

In the multivalued-treatment case, `tebalance summarize` produces output grouped by treatment level. In the [Cattaneo \(2010\)](#) extract, the variable `msmoke` is an ordered categorical variable specifying the number of cigarettes smoked. We begin by tabulating `msmoke`.

```
. tabulate msmoke
```

Cigarettes smoked during pregnancy	Freq.	Percent	Cum.
0 daily	3,778	81.39	81.39
1-5 daily	200	4.31	85.70
6-10 daily	337	7.26	92.96
11+ daily	327	7.04	100.00
Total	4,642	100.00	

All the treatment groups have significantly smaller numbers of observations than the control group of not smoking. Still, each group has at least 200 observations. We continue by quietly fitting a candidate IPW model and reporting the baseline summaries.

```
. quietly teffects ipw (bweight) (msmoke mmarried mage prenatal1 fbaby)
. tebalance summarize, baseline
Covariate balance summary
```

Treatment	Observations	
	Raw	Weighted
0 daily =	3,778	1,164.8
1-5 daily =	200	1,164.4
6-10 daily =	337	1,157.9
11+ daily =	327	1,154.9
Total =	4,642	4,642.0

	Means		Variances	
	Control	Treated	Control	Treated
1-5 daily				
mmarried	.7514558	.455	.1868194	.2492211
mage	26.81048	24.64	31.87141	31.44764
prenatal1	.8268925	.695	.1431792	.2130402
fbaby	.4531498	.48	.2478707	.2508543
6-10 daily				
mmarried	.7514558	.4480712	.1868194	.2480394
mage	26.81048	25.06231	31.87141	27.07051
prenatal1	.8268925	.6795252	.1431792	.2184188
fbaby	.4531498	.3827893	.2478707	.2369648
11+ daily				
mmarried	.7514558	.5107034	.1868194	.250652
mage	26.81048	25.59633	31.87141	26.93471
prenatal1	.8268925	.6972477	.1431792	.2117409
fbaby	.4531498	.293578	.2478707	.2080261

The results for the control level of 0 daily are repeated for the treatment group. These results give a sense of the scale of imbalance in the raw data. Now we compute the balance statistics.

```
. tebalance summarize
```

```
Covariate balance summary
```

Treatment	Observations	
	Raw	Weighted
0 daily =	3,778	1,164.8
1-5 daily =	200	1,164.4
6-10 daily =	337	1,157.9
11+ daily =	327	1,154.9
Total =	4,642	4,642.0

	Standardized differences		Variance ratio	
	Raw	Weighted	Raw	Weighted
1-5 daily				
mmarried	-.634909	-.0016208	1.334021	1.001406
mage	-.3857482	-.0219656	.9867038	.9905584
prenatal1	-.312519	-.0012611	1.487927	1.001898
fbaby	.053769	.0422102	1.012037	1.008631
6-10 daily				
mmarried	-.6506304	-.0108454	1.327696	1.009331
mage	-.3220222	-.0836571	.8493666	.7984901
prenatal1	-.3465797	-.0100232	1.525493	1.015051
fbaby	-.1429048	.0268118	.9560018	1.005899
11+ daily				
mmarried	-.5147672	-.0212969	1.34168	1.018136
mage	-.2239116	-.0636951	.8451058	.8468934
prenatal1	-.3077549	-.0380744	1.478852	1.056645
fbaby	-.3342243	.0155427	.8392526	1.003598

These results indicate that the IPW estimator probably did not fully balance the covariates (the variance ratios for `mage` at the daily levels of 6–10 cigarettes and 11-plus cigarettes are not close to 1). At this point, we would use a richer model and see whether it balanced the covariates.

Note that we cannot use `tebalance override`, because it has not been implemented for multivalued treatments.

◀

## Stored results

`tebalance summarize` stores the following in `r()`:

Matrices

`r(size)`      number of observations in the raw and matched or weighted samples  
`r(table)`     table of covariate statistics

## Reference

Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.



## Also see

- [TE] [stteffects intro](#) — Introduction to treatment effects for observational survival-time data
- [TE] [stteffects ipw](#) — Survival-time inverse-probability weighting
- [TE] [stteffects ipwra](#) — Survival-time inverse-probability-weighted regression adjustment
- [TE] [tebalance](#) — Check balance after teffects or stteffects estimation
- [TE] [teffects aipw](#) — Augmented inverse-probability weighting
- [TE] [teffects ipw](#) — Inverse-probability weighting
- [TE] [teffects ipwra](#) — Inverse-probability-weighted regression adjustment
- [TE] [teffects nnmatch](#) — Nearest-neighbor matching
- [TE] [teffects psmatch](#) — Propensity-score matching
- [TE] [teoverlap](#) — Overlap plots

# Title

**teffects** — Treatment-effects estimation for observational data

[Description](#)   [Syntax](#)   [Also see](#)

## Description

`teffects` estimates potential-outcome means (POMs), average treatment effects (ATEs), and average treatment effects on the treated (ATETs) using observational data. Regression-adjustment, inverse-probability-weighting, and matching estimators are provided, as are doubly robust methods that combine regression adjustment and inverse-probability weighting.

The outcomes can be continuous, binary, count, fractional, or nonnegative. The treatment model can be binary, or it can be multinomial, allowing for multivalued treatments.

For a brief description and example of each estimator, see [Remarks and examples](#) in [TE] [teffects intro](#).

## Syntax

```
teffects subcommand ... [ , options ]
```

<i>subcommand</i>	Description
<code>aipw</code>	augmented inverse-probability weighting
<code>ipw</code>	inverse-probability weighting
<code>ipwra</code>	inverse-probability-weighted regression adjustment
<code>nnmatch</code>	nearest-neighbor matching
<code>psmatch</code>	propensity-score matching
<code>ra</code>	regression adjustment

## Also see

[TE] [teffects intro](#) — Introduction to treatment effects for observational data

[TE] [teffects intro advanced](#) — Advanced introduction to treatment effects for observational data

[TE] [teffects multivalued](#) — Multivalued treatment effects

# Title

**teffects intro** — Introduction to treatment effects for observational data

[Description](#)

[Remarks and examples](#)

[References](#)

[Also see](#)

## Description

This entry provides a nontechnical introduction to treatment-effects estimators and the `teffects` command in Stata. Advanced users may want to instead read [\[TE\] teffects intro advanced](#) or skip to the individual commands' entries.

The `teffects` command estimates average treatment effects (ATEs), average treatment effects among treated subjects (ATETs), and potential-outcome means (POMs) using observational data.

Treatment effects can be estimated using regression adjustment (RA), inverse-probability weights (IPW), and “doubly robust” methods, including inverse-probability-weighted regression adjustment (IPWRA) and augmented inverse-probability weights (AIPW), and via matching on the propensity score or nearest neighbors.

The outcome can be continuous, binary, count, fractional, or nonnegative. Treatments can be binary or multivalued.

## Remarks and examples

This entry presents a nontechnical overview of treatment-effects estimators for those who are new to the subject of treatment-effects estimation or are at least new to Stata's facilities for estimating treatment effects. More advanced users may want to instead read [\[TE\] teffects intro advanced](#) or skip to the individual commands' entries.

Remarks are presented under the following headings:

*[Introduction](#)*

*[Defining treatment effects](#)*

*[Estimating treatment effects](#)*

*[Regression adjustment](#)*

*[Inverse-probability weighting](#)*

*[Doubly robust combinations of RA and IPW](#)*

*[Matching](#)*

*[Caveats and assumptions](#)*

*[A quick tour of the estimators](#)*

*[RA](#)*

*[IPW](#)*

*[IPWRA](#)*

*[AIPW](#)*

*[Nearest-neighbor matching](#)*

*[Propensity-score matching](#)*

*[Video examples](#)*

## Introduction

Suppose we have observed a sample of subjects, some of whom received a treatment and the rest of whom did not. As the name suggests, in most applications, the “subjects” are indeed people. A “treatment” could indeed be a medical treatment such as a new drug regimen or surgical procedure. In social science applications, a treatment could be participation in a job-training program or inclusion in a classroom or school in which a new pedagogical method is being used. However, not all applications use individuals as the subjects. For example, a policy analyst might be interested in examining the impact of an experimental program in which a national agency held a lottery to award only some local governments the resources needed to implement the program. Here the subjects are the local governments, and treatment refers to whether a local government received the resources needed to implement the program.

We would like to know if a treatment has an effect on an outcome  $Y$ . The outcome could be the cholesterol level of a patient taking either an existing statin or a new experimental drug, or the outcome could be the wage offered to a person who either did or did not participate in a job-training program. In an ideal world, we would observe  $Y$  when a subject is treated (which we denote as  $Y_1$ ), and we would observe  $Y$  when the same subject is not treated (which we denote as  $Y_0$ ). We would be careful to make both observations under identical conditions so that the only difference is the presence or absence of the treatment. We could then average the difference between  $Y_1$  and  $Y_0$  across all the subjects in our dataset to obtain a measure of the average impact of the treatment.

Unfortunately, this ideal experiment is almost never available in observational data because it is not possible to observe a specific subject having received the treatment and having not received the treatment. When the outcome is the birthweight of a specific baby and the treatment is the mother smoking while pregnant, it is impossible to observe the baby’s birthweight under both treatments of the mother smoking and the mother not smoking.

A classic solution to this problem is to randomize the treatment. High costs or ethical issues rule out this solution in many observational datasets. For example, we could not ask a random selection of pregnant women to smoke.

The defining characteristic of observational data is that treatment status is not randomized. Moreover, that implies that the outcome and treatment are not necessarily independent. The goal of the estimators implemented by `teffects` is to utilize covariates to make treatment and outcome independent once we condition on those covariates.

The treatment-effect estimators implemented by `teffects` allow us to estimate the efficacy of treatments using observational data. The rest of this entry discusses these treatment-effect estimators at an introductory level. For a more technical introduction, see [\[TE\] `teffects intro advanced`](#).

## Defining treatment effects

We introduce treatment effects more formally by using the potential-outcomes framework, which is also known as the counterfactual framework. What is a potential outcome? Consider a subject that did not receive treatment so that we observe  $Y_0$ . What would  $Y_1$  be for that same subject if it were exposed to treatment? We call  $Y_1$  the potential outcome or counterfactual for that subject. For a subject that did receive treatment, we observe  $Y_1$ , so  $Y_0$  would be the counterfactual outcome for that subject. We can view this as a missing-data problem, and treatment-effect methods can account for that problem.

Treatment-effect estimators allow us to estimate three parameters. The potential-outcome means (POMs) are the means of  $Y_1$  and  $Y_0$  in the population. The average treatment effect (ATE) is the mean of the difference ( $Y_1 - Y_0$ ). Finally, the average treatment effect on the treated (ATET) is the mean of the difference ( $Y_1 - Y_0$ ) among the subjects that actually receive the treatment.

To develop our intuition, suppose we have observed a sample of patients, some of whom received a medication to reduce their blood pressure. Figure 1 plots each of our patient's systolic blood pressures as a function of weight. We use the color green to indicate patients who did not receive the drug and blue to indicate patients who did receive the drug.

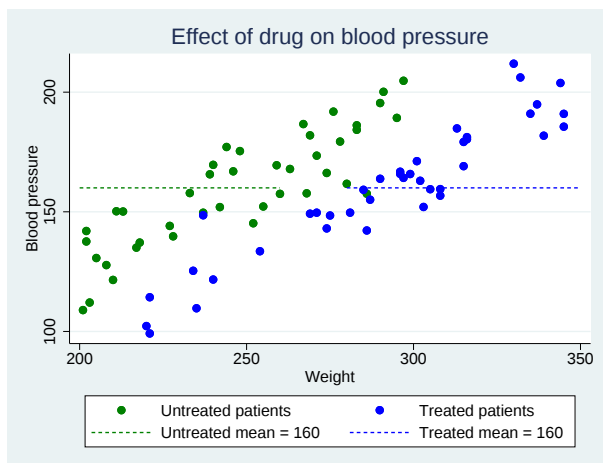


Figure 1

A remarkable feature of our data is that the average blood pressure of patients not taking the drug is 160, and the average blood pressure of patients taking the drug is also 160. Can we therefore conclude that taking the drug has no impact on blood pressure? The answer is no.

Because this is observational data, we could not randomly assign who would receive the drug and who would not. As a result, treatment status could be related to covariates that also affect blood pressure. Heavier patients were more likely to be prescribed the medication, and blood pressure is correlated with weight. The difference in sample means does not estimate the true average treatment effect, because blood pressure depends on weight and weight is correlated with the treatment.

Suppose that we did in fact observe both potential outcomes for all patients. In figure 2, we continue to use solid dots for our observed data points, and we introduce hollow dots to represent the counterfactual outcomes. That is, the green hollow dots represent the blood pressures we would measure if only our treated patients had not taken the drug, and the blue hollow dots represent the blood pressures we would measure if only our untreated patients had taken the drug. The green and blue dashed lines represent the untreated and treated POMs, respectively. That is, the green line represents the mean of all the green dots, and the blue line represents the mean of all the blue dots.

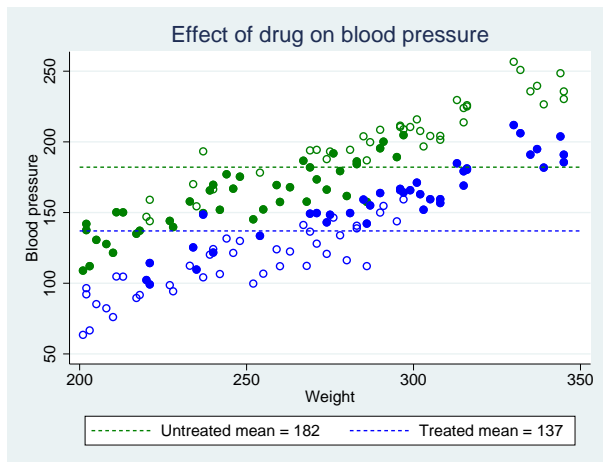


Figure 2

If we did have the data represented by the hollow dots, then we could say that the average treatment effect is the difference between the mean of all the green dots and the mean of all the blue dots. In this ideal scenario, there are no missing data on the other potential outcome, and we have all the data we need to use the difference in means to estimate the ATE.

Looking at figure 2, we can see why a difference in means using only the solid dots does not estimate the ATE. Using only the solid green dots underestimates the average blood pressure for untreated individuals, and using only the solid blue dots overestimates the average blood pressure for treated individuals.

Estimating an ATE is essentially a missing-data problem. When covariates that affect the potential outcomes are related to treatment, we cannot use a difference in sample means, because the missing data are informative.

The treatment-effect estimators implemented in `teffects` allow for covariates like weight to be related to the potential outcomes and the treatment. Essentially, the estimators implemented by `teffects` utilize covariates to fill in the hollow circles or otherwise account for how the missing data depend on covariates that affect the potential outcomes.

## Estimating treatment effects

We cannot estimate the ATE by simply taking the difference between the sample means for the treated and untreated subjects, because there are covariates that are related to the potential outcomes and the treatment. The estimators implemented by `teffects` require us to specify enough of these covariates so that after we condition on these covariates, any remaining influences on the treatment are not related to the potential outcomes. `teffects` implements several different estimators to accomplish this, including regression adjustment (RA), inverse-probability weighting (IPW), “doubly robust” methods that combine elements of RA and IPW, and matching methods. Here we introduce the methods by using intuition and simple examples.

See [\[TE\] `teffects` intro advanced](#) for a more technical introduction, and see the individual commands’ entries for estimator-specific details.

## Regression adjustment

The RA method extends the idea of using sample means to estimate treatment effects by using a regression model to predict potential outcomes adjusted for covariates. In the examples here, we use linear regression, but the `teffects ra` command provides you with the flexibility to use logistic, probit, and heteroskedastic probit regression models for binary outcomes as well as Poisson regression for nonnegative outcomes; see [TE] [teffects ra](#) for more information.

`bweightex.dta` is a hypothetical dataset based on [Cattaneo \(2010\)](#) that we have created to illustrate treatment-effects estimators using graphs. The subjects in this dataset are women who were pregnant, some of whom smoked during the pregnancy. The outcome variable is the birthweight of the baby, and we want to know whether smoking during pregnancy affects the birthweight. The dataset also contains other demographic variables that we will use later.

Figure 3 illustrates the relationship between birthweight and smoking status as a function of the mother's age:

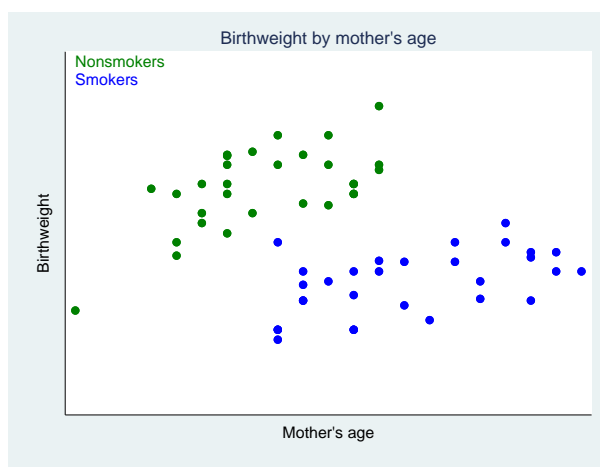


Figure 3

We see that smokers tend to be older than nonsmokers and that birthweight depends on smoking. Therefore, the difference between the sample means of birthweights of babies born to smokers and nonsmokers will not estimate the true average treatment effect.

We also still have the same problem as in the previous section: we do not observe the counterfactual birthweights of babies. Suppose, however, that we did. In [figure 4](#), we use solid points to represent observed birthweights and the colors green to represent nonsmokers and blue to represent smokers. The hollow points represent the counterfactual birthweights. The hollow blue points represent the birthweights of babies that we would observe if only our young nonsmoking mothers had instead smoked during their pregnancies. Similarly, the hollow green points represent the birthweights of babies that we would observe if only our older smoking mothers had instead not smoked during their pregnancies.

Figure 4 suggests a way to estimate the potential outcomes for each mother:

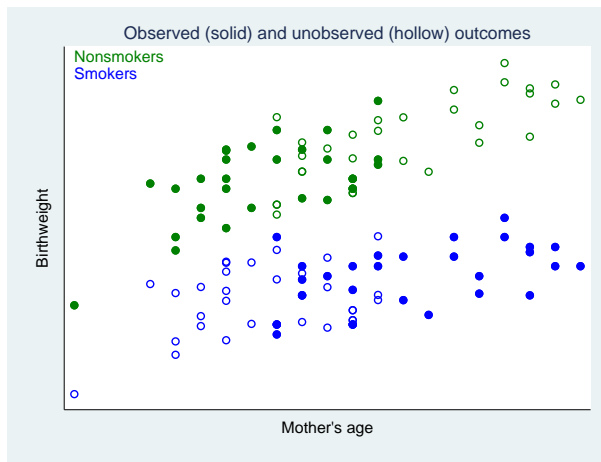


Figure 4

We could fit a linear regression of birthweight on mother's age by using the observed birthweights for nonsmokers, and we could do likewise for smokers. The following graph includes these two regression lines:

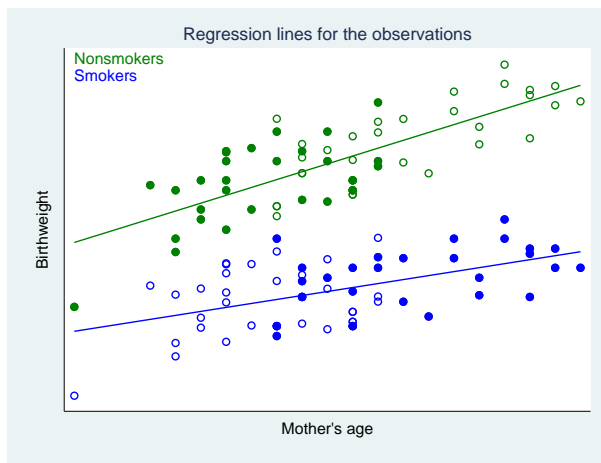


Figure 5

Figure 5 illustrates the principle behind the RA method. We use the green regression line to predict each baby's birthweight assuming the mother did not smoke, and we use the blue regression line to predict each baby's birthweight assuming the mother did smoke. The treatment effect of smoking for a mother of a particular age is the vertical difference between the green and blue regression lines.

The three parameters we mentioned in the introduction are now easy to estimate. For each mother, we obtain two values, say,  $bw_0$  and  $bw_1$ , representing our predictions of her baby's birthweight assuming the mother did not or did smoke, respectively. The means of these variables represent the untreated and treated POMs. The ATE is the sample mean of the difference ( $bw_1 - bw_0$ ), and the ATET is the sample mean of that difference computed using only the mothers who in fact did smoke during pregnancy.



Adding the circles highlights the fact that the average age is higher for smokers than for nonsmokers. Even though the blue and green lines have different slopes, if the average age was the same for smokers and nonsmokers, a difference in the sample means of birthweights could still estimate the true ATE.

Figure 5 lets us address one more issue. Users who are versed in regression analysis may be inclined to estimate the effect of smoking using a regression model for birthweight as a function of smoking and the mother's age. We clearly see in figure 5 that regression lines for smokers and nonsmokers have different slopes—the effect of age on birthweight is not the same for smokers and nonsmokers. In regression analysis, we would therefore include an interaction term between smoking and age. The RA method fits separate regression lines for smokers and nonsmokers, which also handles these differential effects of age on smoking.

## Inverse-probability weighting

As we remarked in our discussion of the RA method, we cannot simply use the sample mean birthweights of babies born to smokers and nonsmokers to estimate the effect of smoking. If we did that, we would conflate the negative effect of smoking with the positive effect of age and the positive relationship between age and smoking. IPW is a treatment-effects estimator that uses weighted means rather than simple unweighted means to disentangle the effects of treatment and other confounders like age.

The concept underlying IPW can be gleaned from figure 2, where, as you will recall, the hollow points represent counterfactual outcomes. As we demonstrated in *Defining treatment effects*, we could estimate the average treatment effect if we knew the means of all the nonsmoking outcomes and the means of all the smoking outcomes. In the context of figure 4, we need the mean of all the green points, both solid and hollow, and the mean of all the blue.

If we could observe all of these points, then the ATE would be the difference between those two means. However, the outcomes illustrated by the hollow circles are unobserved. IPW estimators view the hollow circles as missing data and use weights to correct the estimates of the treated and untreated sample means for the missing data. If we calculate the mean nonsmoking birthweight using just the solid green points, that mean is biased downward because we are ignoring the hollow green points, which correspond to higher birthweights.

In IPW, we apply more weight to the solid green points corresponding to older mothers and less weight to those corresponding to younger mothers. Using this weighting scheme will pull up the estimated mean birthweight of babies born to nonsmoking mothers to estimate the true mean of all nonsmoking outcomes. The method for obtaining the mean smoking birthweight is virtually the same: we need to apply more weight to the younger smoking mothers than to the older smoking mothers to better approximate the true mean of all smoking outcomes.

Where do these weights for the weighted means come from? As the name implies, IPW uses the inverse (reciprocal) of the probability of being in the observed treatment group. These probabilities are obtained by modeling the observed treatment as a function of subject characteristics that determine treatment group. In our exposition of the RA method, we focused solely on the mother's age and smoking status as determinants of each baby's birthweight. To make the results comparable, we will use the same model in this example.

We first fit a logistic model of the mother's smoking status, `mbsmoke`, as a function of the mother's age (`mage`):

```
. use https://www.stata-press.com/data/r17/bweightex
(Hypothetical birthweight data)
. logistic mbsmoke mage
Logistic regression
Log likelihood = -26.362201
Number of obs = 60
LR chi2(1) = 30.45
Prob > chi2 = 0.0000
Pseudo R2 = 0.3661
```

mbsmoke	Odds ratio	Std. err.	z	P> z	[95% conf. interval]
mage	1.631606	.21316	3.75	0.000	1.263022 2.107754
_cons	7.76e-06	.0000243	-3.76	0.000	1.69e-08 .0035718

Note: `_cons` estimates baseline odds.

Next, we compute the inverse-probability weights, which we will store in a variable called `ps`. In the IPW method, for subjects who did receive treatment, the weight is equal to the reciprocal of the predicted probability of treatment. For subjects who did not receive treatment, the weight is equal to the reciprocal of the predicted probability of not receiving treatment; the probability of not receiving treatment is just one minus the probability of receiving treatment:

```
. predict ps
(option pr assumed; Pr(mbsmoke))
. replace ps = 1/ps if mbsmoke==1
(30 real changes made)
. replace ps = 1/(1-ps) if mbsmoke==0
(30 real changes made)
```

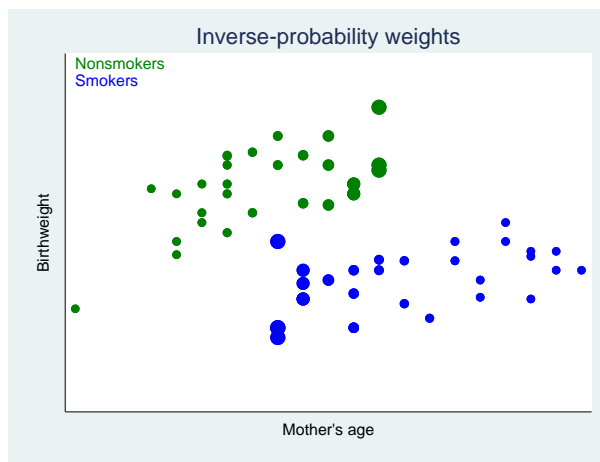


Figure 6

Figure 6 replicates [figure 3](#) with one twist. Rather than making all the points the same size, we have made the size of the points proportional to the IPW variable `ps`. Notice that the largest blue points correspond to the youngest smoking mothers in our sample, so they will receive the most weight when we compute the weighted mean birthweight of babies born to smoking mothers, just as we explained we wanted to do. Similarly, the green points corresponding to older nonsmoking mothers are larger, representing larger weights.

There is a caveat to the IPW estimator. When we fit our logistic or probit model to obtain the predicted probabilities, it is possible that some of the predictions will be close to zero. Because the IPW is the reciprocal of that probability, the weight becomes arbitrarily large as the probability goes to zero. In those cases, the IPW can become unstable. We can improve the estimated IPW by developing a more accurate treatment model. For example, in our dataset, we have other variables such as marital status and the education level of the baby's father that may also help predict whether the mother smoked during pregnancy. We excluded these variables for simplicity, but in a real analysis, we would want to use all relevant data.

This phenomenon of unstable IPWs is related to the concept of overlap, which means that every subject must have a strictly positive probability of obtaining treatment. We remarked that in our sample, we had few young mothers who smoked. As should be clear from [figure 6](#), the overlap assumption is likely to be violated—young mothers do not appear to have a positive probability of being smokers. We would want to check this assumption before proceeding with an IPW analysis. See [\[TE\] teoverlap](#) and [\[TE\] teffects intro advanced](#) for more information about overlap.

Another limitation of the IPW estimator is that we are using weighted means to estimate the POMs and ATE. Thus, unlike the RA estimator, we cannot obtain subject-level predictions of the treatment effects or potential outcomes, because we do not have the two regression lines that we can use to predict outcomes for each subject.

## Doubly robust combinations of RA and IPW

You may have noticed a clear distinction between the RA and IPW estimators. In the case of RA, we built linear regression models to predict the outcomes (birthweights) of each subject but said nothing about how treatment (smoking) arises. In the case of IPW, we built a logistic regression model to predict treatment status but did not build a formal model of the outcome. Doubly robust estimators combine the outcome modeling strategy of RA and the treatment modeling strategy of IPW. These estimators have a remarkable property: although they require us to build two models, we only need to specify one of the two models correctly. If we misspecify the treatment model but correctly specify the outcome model, we still obtain correct estimates of the treatment effect. If we correctly specify the treatment model but misspecify the outcome model, we again will obtain correct estimates of the treatment effect.

Stata's `teffects` command implements two doubly robust estimators, the augmented inverse-probability-weighted (AIPW) estimator and the inverse-probability-weighted regression-adjustment (IPWRA) estimator. These estimators combine elements of RA and IPW to be more robust to misspecification.

The AIPW estimator is an IPW estimator that includes an augmentation term that corrects the estimator when the treatment model is misspecified. When the treatment model is correctly specified, the augmentation term vanishes as the sample size becomes large. Like the IPW, the AIPW does not perform well when the predicted treatment probabilities are too close to zero or one.

The IPWRA estimator is an RA estimator that uses estimated inverse-probability weights to correct the estimator when the regression function is misspecified. When the regression function is correctly specified, the weights do not affect the consistency of the estimator.

## Matching

Matching estimators are based on the idea of comparing the outcomes of subjects that are as similar as possible with the sole exception of their treatment status. In our birthweight and smoking example, we could select a mother who smokes and select a mother of the same age who does not smoke and compare the birthweights of their infants. The data of each mother serve as the potential outcome for the other mother.

For a single covariate such as age, identifying a pair of comparable mothers is not difficult. If we have a second covariate that is categorical, such as race, we might still be able to identify pairs of mothers who are the same age and of the same race assuming our dataset is large enough. However, once we consider covariates that are measured on continuous scales or allow for more than a few discrete ones, then finding identical matches is a challenge. The solution is to use what is called a similarity measure, which is a statistic that measures how “close” two observations are. **teffects** offers two methods to find comparable observations based on similarity measures: nearest-neighbor matching and propensity-score matching.

Nearest-neighbor matching (NNM) is accomplished by calculating the “distance” between pairs of observations with regard to a set of covariates and then “matching” each subject to comparable observations that are closest to it. For example, suppose we have a variable that records each subject’s annual income to the penny. Say one subject who received treatment had an income of \$69,234.21. The likelihood that our dataset has an untreated subject who also earned \$69,234.21 is nil. However, we can determine the difference between each untreated subject’s income and our treated subject’s income, then match our treated subject with the untreated subjects whose income differences are smallest. Measuring the distance between subjects when we have multiple covariates is no challenge. By default, **teffects** uses what is known as the Mahalanobis distance, which is really nothing more than the Pythagorean theorem adapted to handle the fact that covariates may be correlated and measured on different scales.

NNM does not use a formal model for either the outcome or the treatment status, but this flexibility comes at a price. When matching on more than one continuous covariate, the NNM estimator must be augmented with a bias-correction term. **teffects** `nnmatch` uses a linear function of the covariates specified in the `biasadj()` option to remove the large-sample bias.

Propensity-score matching (PSM) is an alternative to NNM. PSM matches on the estimated predicted probabilities of treatment, known as the propensity scores. PSM does not require bias correction, because it uses a model for the treatment. If the treatment model is reasonably well specified, PSM will perform at least as well as NNM; see [TE] **teffects intro advanced**.

## Caveats and assumptions

To use the estimators implemented in **teffects**, we must make several assumptions about the process that generated our data. Different estimators and statistics may require slightly more or slightly less restrictive assumptions and may exhibit varying degrees of robustness to departures from these assumptions, but in general, all the estimators require some form of the following three assumptions.

The independent and identically distributed (i.i.d.) sampling assumption ensures that the outcome and treatment status of each individual are unrelated to the outcome and treatment status of all the other individuals in the population. Correlated data arising from hierarchical or longitudinal study designs do not meet this assumption.

The conditional-independence (CI) assumption means once we control for all observable variables, the potential outcomes are independent of treatment assignment. The easiest way to understand the CI assumption is to understand when it is violated. In our birthweight example, suppose mothers

who did not smoke were more health conscious and consumed better prenatal diets than those who did smoke. Unless we explicitly controlled for health awareness or diet, our model would violate the CI assumption: the mother's decision to smoke or not smoke would not be independent of the baby's birthweight. If we did not control for health awareness, we would overstate the negative impact of smoking on birthweight. Babies born to mothers who smoke weigh less than babies born to nonsmoking mothers not just because of the effects of cigarettes but also because of poorer prenatal diets.

In a study examining the effect of a job-training program, the CI assumption requires that there not be any unobserved factors such as ambition or work ethic that influence both whether a person enrolls in the program and the wage received upon completion. To use the methods implemented by the `teffects` estimators, we must have variables in our dataset that allow us to control for those types of factors.

We mentioned the third assumption, overlap, in our discussions of IPW. More formally, the overlap assumption states that each individual have a positive probability of receiving treatment. In our birthweight example, we noted that there were no observations on young smokers and older nonsmokers. Perhaps we just have an unlucky sample, but to accurately assess the impact of treatment using these methods, we must have overlap to accurately estimate the counterfactual birthweights. In the context of matching estimators, overlap essentially means that we can actually match treated subjects with similar nontreated subjects.

## A quick tour of the estimators

The `teffects` command implements six estimators of treatment effects. We introduce each one by showing the basic syntax one would use to apply them to our birthweight example. See each command's entry for more information.

### RA

`teffects ra` implements the RA estimator. We estimate the effect of a mother's smoking behavior (`mbsmoke`) on the birthweight of her child (`bweight`), controlling for marital status (`mmarried`), the mother's age (`mage`), whether the mother had a prenatal doctor's visit in the baby's first trimester (`prenatal1`), and whether this baby is the mother's first child (`fbaby`). We use linear regression (the default) to model `bweight`:

```
. use https://www.stata-press.com/data/r17/cattaneo2
. teffects ra (bweight mmarried mage prenatal1 fbaby) (mbsmoke)
```

### IPW

`teffects ipw` implements the IPW estimator. Here we estimate the effect of smoking by using a probit model to predict the mother's smoking behavior as a function of marital status, the mother's age, and indicators for first-trimester doctor's visits and firstborn status:

```
. teffects ipw (bweight) (mbsmoke mmarried mage prenatal1 fbaby, probit)
```

## IPWRA

`teffects ipwra` implements the IPWRA estimator. We model the outcome, birthweight, as a linear function of marital status, the mother's age, and indicators for first-trimester doctor's visits and firstborn status. We use a logistic model (the default) to predict the mother's smoking behavior, using the same covariates as explanatory variables:

```
. teffects ipwra (bweight mmarried mage prenatal1 fbaby) ///  
                (mbsmoke mmarried mage prenatal1 fbaby)
```

## AIPW

`teffects aipw` implements the AIPW estimator. Here we use the same outcome- and treatment-model specifications as we did with the IPWRA estimator:

```
. teffects aipw (bweight mmarried mage prenatal1 fbaby) ///  
                (mbsmoke mmarried mage prenatal1 fbaby)
```

## Nearest-neighbor matching

`teffects nnmatch` implements the NNM estimator. In this example, we match treated and untreated subjects based on marital status, the mother's age, the father's age, and indicators for first-trimester doctor's visits and firstborn status. We use the Mahalanobis distance based on the mother's and father's ages to find matches. We use exact matching on the other three variables to enforce the requirement that treated subjects are matched with untreated subjects who have the same marital status and indicators for first-trimester doctor's visits and firstborn statuses. Because we are matching on two continuous covariates, we request that `teffects nnmatch` include a bias-correction term based on those two covariates:

```
. teffects nnmatch (bweight mage fage) (mbsmoke),          ///  
                  ematch(prenatal1 mmarried fbaby) biasadj(mage fage)
```

## Propensity-score matching

`teffects psmatch` implements the PSM estimator. Here we model the propensity score using a probit model, incorporating marital status, the mother's age, and indicators for first-trimester doctor's visits and firstborn status as covariates:

```
. teffects psmatch (bweight) (mbsmoke mmarried mage prenatal1 fbaby, probit)
```

## Video examples

[Introduction to treatment effects in Stata, part 1](#)

[Introduction to treatment effects in Stata, part 2](#)

## References

- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cattaneo, M. D., R. Titiunik, and G. Vazquez-Bare. 2020. Analysis of regression-discontinuity designs with multiple cutoffs or multiple scores. *Stata Journal* 20: 866–891.
- Cerulli, G. 2017. Identification and estimation of treatment effects in the presence of (correlated) neighborhood interactions: Model and Stata implementation via `ntreatreg`. *Stata Journal* 17: 803–833.
- . 2020. Nonparametric synthetic control using the `npsynth` command. *Stata Journal* 20: 844–865.
- Cerulli, G., and M. Ventura. 2019. Estimation of pre- and posttreatment average treatment effects with binary time-varying treatment using Stata. *Stata Journal* 19: 551–565.
- Drukker, D. M. 2016. A generalized regression-adjustment estimator for average treatment effects from panel data. *Stata Journal* 16: 826–836.
- Galiani, S., and B. Quistorff. 2017. The `synth_runner` package: Utilities to automate synthetic control estimation using `synth`. *Stata Journal* 17: 834–849.
- Heß, S. 2017. Randomization inference with Stata: A guide and software. *Stata Journal* 17: 630–651.
- Linden, A., C. Huber, and G. T. Wodtke. 2021. A regression-with-residuals method for analyzing causal mediation: The `rwrmcd` package. *Stata Journal* 21: 559–574.
- Pierce, M., and R. Emsley. 2021. Estimating and evaluating personalized treatment recommendations from randomized trials with `ptr`. *Stata Journal* 21: 348–359.

## Also see

[TE] [teffects](#) — Treatment-effects estimation for observational data

[TE] [teffects intro advanced](#) — Advanced introduction to treatment effects for observational data

[TE] [teffects multivalued](#) — Multivalued treatment effects

# Title

**teffects intro advanced** — Advanced introduction to treatment effects for observational data

Description  
Also see

Remarks and examples

Acknowledgments

References

## Description

This entry provides a technical overview of treatment-effects estimators and their implementation in Stata. Those who are new to treatment-effects estimation may want to instead see [TE] [teffects intro](#).

The `teffects` command estimates average treatment effects (ATEs), average treatment effects among treated subjects (ATETs), and potential-outcome means (POMs) using observational data.

Treatment effects can be estimated using regression adjustment (RA), inverse-probability weights (IPW), and “doubly robust” methods, including inverse-probability-weighted regression adjustment (IPWRA) and augmented inverse-probability weights (AIPW), and via matching on the propensity score or nearest neighbors.

The outcome can be continuous, binary, count, fractional, or nonnegative. Treatments can be binary or multivalued.

## Remarks and examples

This entry presents a technical overview of treatment-effects estimators and their implementation in Stata. Users who are new to treatment-effects estimators for observational data should instead read [TE] [teffects intro](#).

Remarks are presented under the following headings:

- Introduction*
- Defining treatment effects*
- The potential-outcome model*
- Assumptions needed for estimation*
  - The CI assumption*
  - The overlap assumption*
  - The i.i.d. assumption*
- Comparing the ATE and ATET*
- Overview of treatment-effect estimators*
- RA estimators*
- IPW estimators*
- AIPW estimators*
- IPWRA estimators*
- Nearest-neighbor matching estimators*
- Propensity-score matching estimators*
- Choosing among estimators*
- Model choice*



## Introduction

The `teffects` commands estimate treatment effects from observed data. A treatment effect is the change in an outcome caused by a subject, often an individual, getting one treatment instead of another. We cannot estimate individual-level treatment effects, because we only observe each individual getting one or another treatment.

Potential-outcome models provide a solution to this missing-data problem and allow us to estimate the distribution of individual-level treatment effects. A potential-outcome model specifies the potential outcomes that each individual would obtain under each treatment level, the treatment assignment process, and the dependence of the potential outcomes on the treatment assignment process.

When the potential outcomes do not depend on the treatment levels, after conditioning on covariates, regression estimators, inverse-probability-weighted estimators, and matching estimators are commonly used.

What we call the potential-outcome model is also known as the Rubin causal model and the counterfactual model. See [Rubin \(1974\)](#); [Holland \(1986\)](#); [Robins \(1986\)](#); [Heckman \(1997\)](#); [Heckman and Navarro-Lozano \(2004\)](#); [Imbens \(2004\)](#); [Cameron and Trivedi \(2005, chap. 2.7\)](#); [Imbens and Wooldridge \(2009\)](#); and [Wooldridge \(2010, chap. 21\)](#) for more detailed discussions.

For a discussion related to health econometrics, see [Deb, Norton, and Manning \(2017\)](#).

## Defining treatment effects

Three parameters are often used to measure treatment effects: the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs). In this section, we define each of these terms and introduce the notation and parameters used in the rest of our discussion.

In the binary-treatment case, the two potential outcomes for each individual are  $y_{0i}$  and  $y_{1i}$ ;  $y_{0i}$  is the outcome that would be obtained if  $i$  does not get the treatment, and  $y_{1i}$  is the outcome that would be obtained if  $i$  gets the treatment.  $y_{0i}$  and  $y_{1i}$  are realizations of the random variables  $y_0$  and  $y_1$ . Throughout this entry,  $i$  subscripts denote realizations of the corresponding unsubscripted random variables. We do not discuss multivalued treatments here, because doing so only increases the number of parameters and notation required and detracts from the essential points; see [\[TE\] teffects multivalued](#) for information about multivalued treatments.

The parameters of interest summarize the distribution of the unobservable individual-level treatment effect  $y_1 - y_0$ . In defining the parameters,  $t$  denotes a random treatment,  $t_i$  denotes the treatment received by individual  $i$ ,  $t = 1$  is the treatment level, and  $t = 0$  is the control level. Given this notation, we can now define our parameters of interest.

**ATE** The ATE is the average effect of the treatment in the population:

$$\text{ATE} = E(y_1 - y_0)$$

**POM** The POM for treatment level  $t$  is the average potential outcome for that treatment level:

$$\text{POM}_t = E(y_t)$$

**ATET** The ATET is the average treatment effect among those that receive the treatment:

$$\text{ATET} = E(y_1 - y_0 | t = 1)$$

For an illustration of these concepts, see [Defining treatment effects](#) in [\[TE\] teffects intro](#).

## The potential-outcome model

Next we specify a potential-outcome model that serves as a touchstone for the rest of our discussion. The model described here generates data in which  $y_i$  is the observed outcome variable,  $t_i$  is the treatment variable,  $\mathbf{x}_i$  is a vector of covariates that affect the outcome, and  $\mathbf{w}_i$  is a vector of covariates that affect the treatment assignment.  $\mathbf{x}_i$  and  $\mathbf{w}_i$  may have elements in common.

This potential-outcome model specifies that the observed outcome variable  $y$  is  $y_0$  when  $t = 0$  and that  $y$  is  $y_1$  when  $t = 1$ . Algebraically, we say that

$$y = (1 - t)y_0 + ty_1$$

The functional forms for  $y_0$  and  $y_1$  are

$$y_0 = \mathbf{x}'\beta_0 + \epsilon_0 \tag{1}$$

$$y_1 = \mathbf{x}'\beta_1 + \epsilon_1 \tag{2}$$

where  $\beta_0$  and  $\beta_1$  are coefficients to be estimated, and  $\epsilon_0$  and  $\epsilon_1$  are error terms that are not related to  $\mathbf{x}$  or  $\mathbf{w}$ . This potential-outcome model separates each potential outcome into a predictable component,  $\mathbf{x}\beta_t$ , and an unobservable error term,  $\epsilon_t$ .

The treatment assignment process is

$$t = \begin{cases} 1 & \text{if } \mathbf{w}'\gamma + \eta > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where  $\gamma$  is a coefficient vector, and  $\eta$  is an unobservable error term that is not related to either  $\mathbf{x}$  or  $\mathbf{w}$ . The treatment assignment process is also separated into a predictable component,  $\mathbf{w}'\gamma$ , and an unobservable error term,  $\eta$ .

We emphasize six points about this model:

1. The observed data from this model contain  $y_i$ ,  $t_i$ ,  $\mathbf{w}_i$ , and  $\mathbf{x}_i$ . The data do not reveal both  $y_{0i}$  and  $y_{1i}$  for any given  $i$ .
2. The model for  $t$  determines how the data on  $y_0$  and  $y_1$  are missing.
3. The model separates the potential outcomes and treatment assignment into observable and unobservable components.
4. Whether  $\eta$  is independent of the vector  $(\epsilon_0, \epsilon_1)$  is essential in specifying the set of available estimators.
5. The coefficient vectors  $\beta_0$ ,  $\beta_1$ , and  $\gamma$  are auxiliary parameters. We use estimates of these coefficient vectors to estimate the ATE, the POMs, and the ATET.
6. For notational simplicity, we represented  $y_0$  and  $y_1$  as linear functions. In practice, we can use other functional forms.

In specifying this potential-outcome model, we explicitly showed the functional forms for the potential outcomes and the treatment assignment process. To ease subsequent discussions, we refer to the set of functional forms for the potential outcomes as the “outcome model”, and we refer to the treatment assignment process as the “treatment model”.

## Assumptions needed for estimation

As with any type of estimator, we must make some assumptions to use treatment-effects estimators. The particular assumptions we need for each estimator implemented by `teffects` and for each effect parameter vary, but some version of each of the following is required.

**CI** The conditional-independence CI assumption restricts the dependence between the treatment model and the potential outcomes.

**Overlap** The overlap assumption ensures that each individual could receive any treatment level.

**i.i.d.** The independent and identically distributed (i.i.d.) sampling assumption ensures that the potential outcomes and the treatment status of each individual are unrelated to the potential outcomes and treatment statuses of all other individuals in the population.

We now discuss each assumption in detail.

### The CI assumption

After conditioning on covariates, when no unobservable variable affects both treatment assignment and the potential outcomes, the potential outcomes are conditionally independent of the treatment. In epidemiological jargon, there are no unmeasured confounders. In econometric jargon, we have selection on observables. If we observe enough covariates, the potential outcomes may indeed be conditionally independent of the treatment.

Intuitively, the CI assumption says that only the covariates  $\mathbf{x}$  affect both the treatment and the potential outcomes. Any other factors that affect the treatment must be independent of the potential outcomes, and any other factors that affect the potential outcomes must be independent of the treatment. Formally, the CI assumption states that, conditional on covariates  $\mathbf{x}$ , the treatment  $t$  is independent of the vector of potential outcomes  $(y_0, y_1)'$ .

The CI assumption allows us to estimate the effects by regression-adjustment (RA) methods, inverse-probability-weighting (IPW) methods, methods that combine RA and IPW concepts, and matching methods. The data only reveal information about  $E(y_0|\mathbf{x}, \mathbf{w}, t = 0)$  and  $E(y_1|\mathbf{x}, \mathbf{w}, t = 1)$ , but we are interested in an average of  $E(y_0|\mathbf{x}, \mathbf{w})$  and  $E(y_1|\mathbf{x}, \mathbf{w})$ , where  $\mathbf{x}$  represents the outcome covariates and  $\mathbf{w}$  the treatment-assignment covariates. The CI assumption allows us to estimate  $E(y_0|\mathbf{x}, \mathbf{w})$  and  $E(y_1|\mathbf{x}, \mathbf{w})$  directly from the observations for which  $E(y_0|\mathbf{x}, \mathbf{w}, t = 0)$  and  $E(y_1|\mathbf{x}, \mathbf{w}, t = 1)$ , respectively.

For our potential-outcome model presented in (1) through (3), the CI assumption can be viewed as a set of restrictions on the covariance matrix of the error terms. Suppose that the vector of unobservables  $(\epsilon_0, \epsilon_1, \eta)$  is normally distributed

$$\begin{pmatrix} \epsilon_0 \\ \epsilon_1 \\ \eta \end{pmatrix} \sim N \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_0^2 & \rho_{01}\sigma_0\sigma_1 & \rho_{\eta 0}\sigma_0 \\ \rho_{01}\sigma_0\sigma_1 & \sigma_1^2 & \rho_{\eta 1}\sigma_1 \\ \rho_{\eta 0}\sigma_0 & \rho_{\eta 1}\sigma_1 & 1 \end{pmatrix} \right\} \quad (4)$$

where  $\sigma_0$  is the standard deviation of  $\epsilon_0$ ,  $\rho_{01}$  is the correlation between  $\epsilon_0$  and  $\epsilon_1$ ,  $\sigma_1$  is the standard deviation of  $\epsilon_1$ ,  $\rho_{\eta 0}$  is the correlation between  $\eta$  and  $\epsilon_0$ , and  $\rho_{\eta 1}$  is the correlation between  $\eta$  and  $\epsilon_1$ . As is standard in the normally distributed latent-variable specification of a binary-dependent variable, we normalize the variance of  $\eta$  to 1.

CI specifies that  $\rho_{\eta 0} = \rho_{\eta 1} = 0$  so that we can write (4) as

$$\begin{pmatrix} \epsilon_0 \\ \epsilon_1 \\ \eta \end{pmatrix} \sim N \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_0^2 & \rho_{01}\sigma_0\sigma_1 & 0 \\ \rho_{01}\sigma_0\sigma_1 & \sigma_1^2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right\}$$

Writing the covariance matrix this way makes clear what we mean by conditional independence: unobserved shocks that affect whether a subject is treated do not have any effect on the potential outcomes, and unobserved shocks that affect a potential outcome do not affect treatment.

The command `teffects` implements estimators that require the CI assumption. See [TE] [etregress](#) and [TE] [etpoisson](#) for commands that handle two cases in which the CI assumption is replaced by precise specifications of the joint dependence among the unobservables. [Brown and Mergoupis \(2011\)](#) discuss the `itreatreg` command that extends [TE] [etregress](#).

The CI assumption is also known as unconfoundedness and selection-on-observables in the literature. See [Rosenbaum and Rubin \(1983\)](#); [Heckman \(1997\)](#); [Heckman and Navarro-Lozano \(2004\)](#); [Cameron and Trivedi \(2005, sec. 25.2.1\)](#); [Tsiatis \(2006, sec. 13.3\)](#); [Angrist and Pischke \(2009, chap. 3\)](#); [Imbens and Wooldridge \(2009\)](#); and [Wooldridge \(2010, sec. 21.3\)](#). Some discussions with Stata commands can be found in [Becker and Caliendo \(2007\)](#), [Nichols \(2007\)](#), and [Daniel, De Stavola, and Cousens \(2011\)](#).

## □ Technical note

In fact, full CI is stronger than what we need to estimate the ATE, the ATET, or the POMS. For the estimators implemented in `teffects`, we only need a conditional mean independence (CMI) assumption. Intuitively, the CMI assumption says that after accounting for the covariates  $\mathbf{x}_i$ , the treatment does not affect the conditional mean of each potential outcome. Formally, the CMI requires that  $E(y_0|\mathbf{x}, t) = E(y_0|\mathbf{x})$  and that  $E(y_1|\mathbf{x}, t) = E(y_1|\mathbf{x})$ . The CMI assumption allows the conditional variance to depend on the treatment, while the CI assumption does not.

The CI assumption implies the CMI assumption, but not vice versa.

See [Wooldridge \(2010, sec. 21.2 and 21.3\)](#) for an excellent introduction to this topic, and see [Cattaneo, Drukker, and Holland \(2013\)](#) for some discussion of the multiple treatment case. □

## The overlap assumption

The overlap assumption requires that each individual have a positive probability of receiving each treatment level. Formally, the overlap assumption requires that for each possible  $\mathbf{x}$  in the population and each treatment level  $\tilde{t}$ ,  $0 < \Pr(t = \tilde{t}|\mathbf{x}) < 1$ . [Rosenbaum and Rubin \(1983\)](#) call the combination of the CI and overlap assumptions strong ignorability; see also [Abadie and Imbens \(2006, 237–238\)](#) and [Imbens and Wooldridge \(2009\)](#).

## The i.i.d. assumption

The third of the three assumptions listed above is the i.i.d. assumption; it is the standard assumption of having an i.i.d. sample from the population. In potential-outcome models, i.i.d. sampling implies that the potential outcomes and treatment status of each individual are unrelated to the potential outcomes and treatment statuses of all the other individuals in the population. I.i.d. sampling rules out interactions among the individuals. For instance, models of vaccinations in epidemiology and general

equilibrium effects in economics violate the independence assumption. This third assumption is a part of what is known as the stable unit treatment value assumption (SUTVA); see [Wooldridge \(2010, 905\)](#) and [Imbens and Wooldridge \(2009\)](#).

## Comparing the ATE and ATET

When comparing the ATE and the ATET, two points should be mentioned.

First, the assumptions required to estimate the ATET are less restrictive than the assumptions required to estimate the ATE. Estimating the ATET requires a weaker form of the CI assumption and a weaker version of the overlap assumption.

To estimate the ATE under CI, we require that the unobservables in the treatment model be conditionally independent of the unobservables in both potential outcomes. In contrast, we can estimate the ATET under CI when the unobservables in the treatment model are conditionally independent of just the control-level potential outcome; see [Wooldridge \(2010, 906–912\)](#).

Although the ATE version of overlap requires that all covariate patterns have a positive probability of being allocated to each treatment state, we can estimate the ATET when only the covariate patterns for which someone is treated have a positive probability of being allocated to each treatment state. This weaker overlap assumption can be important in some studies. For example, [Heckman \(1997\)](#) discusses how the ATET makes sense in job-training programs for which many types of individuals have zero chance of signing up. See also [Wooldridge \(2010, 911–913\)](#).

Second, the ATET reduces to the ATE when the mean of the covariates among the treated is the same as the mean of the covariates in the population and when the average contribution from the unobservables for the participants is zero.

## Overview of treatment-effect estimators

We can classify the estimators implemented by `teffects` into five categories: 1) estimators based on a model for the outcome variable; 2) estimators based on a model for treatment assignment; 3) estimators based on models for both the outcome variable and the treatment assignment; 4) estimators that match on covariates; and 5) estimators that match on predicted probabilities of treatment. Within each category of estimator, there is a variety of choices about the functional forms for the models.

Because there are several categories of estimators, the user must decide whether to model the outcomes, the probability of treatment, or both. Under correct model specification, using an outcome model and a model for the probability of treatment will produce more efficient estimates. Surprisingly, some of the estimators that use both models only require that one of the two be correctly specified to consistently estimate the effects of interest, a property known as the double-robust property.

With the exception of using a matching estimator with a single continuous covariate, some choice of functional forms is required. There are two aspects one must consider when choosing the functional form for the outcome or treatment assignment. First, one must select the functional form for the conditional mean or conditional probability; depending on the variable being modeled, a linear, a binary choice, or an exponential model may be appropriate. Second, one must determine the appropriate polynomials of the covariates to include in the model. `teffects` offers a wide variety of options to specify different functional form choices for the conditional mean and conditional probability models. The factor-variable notation in Stata allows us to easily specify the desired polynomial in the covariates.

We now provide some intuition behind each type of estimator.

## RA estimators

RA estimators use means of predicted outcomes for each treatment level to estimate each POM. ATEs and ATETs are differences in estimated POMs.

The CI assumption implies that we can estimate  $E(y_0|\mathbf{x})$  and  $E(y_1|\mathbf{x})$  directly from the observations for which  $t = 0$  and  $t = 1$ , respectively. Regression adjustment fits separate regressions for each treatment level and uses averages of the predicted outcomes over all the data to estimate the POMs. The estimated ATEs are differences in the estimated POMs. The estimated ATETs are averages of the predicted outcomes over the treated observations.

RA is a venerable estimator. See Lane and Nelder (1982); Cameron and Trivedi (2005, chap. 25); Wooldridge (2010, chap. 21); and Vittinghoff et al. (2012, chap. 9). The usefulness of RA has been periodically questioned in the literature because it relies on specifying functional forms for the conditional means and because it requires having sufficient observations of each covariate pattern in each treatment level; see Rubin (1973) for an early salvo. Our experience is that RA is an exceptionally useful base-case estimator. We describe its relative advantages and disadvantages in the course of covering other estimators.

## IPW estimators

IPW estimators use weighted averages of the observed outcome variable to estimate means of the potential outcomes. The weights account for the missing data inherent in the potential-outcome framework. Each weight is the inverse of the estimated probability that an individual receives a treatment level. Outcomes of individuals who receive a likely treatment get a weight close to one. Outcomes of individuals who receive an unlikely treatment get a weight larger than one, potentially much larger.

IPW estimators model the probability of treatment without any assumptions about the functional form for the outcome model. In contrast, RA estimators model the outcome without any assumptions about the functional form for the probability of treatment model.

IPW estimators become extremely unstable as the overlap assumption gets close to being violated. When the overlap assumption is nearly violated, some of the inverse-probability weights become very large, IPW estimators produce erratic estimates, and the large-sample distribution provides a poor approximation to the finite-sample distribution of IPW estimators. This instability occurs even though the functional form for the treatment model is correctly specified.

In contrast, when the overlap assumption is nearly violated, there are very few observations in a treatment level for some covariate patterns, so RA estimators use the model to predict in regions in which there are very little data. If the model is well specified and there are “enough” observations, an RA estimator will not become unstable as quickly as an IPW estimator, and the large-sample distribution of the RA estimator still provides a good approximation to the finite-sample distribution. However, in real situations in which “all models are approximate”, relying on a correctly specified outcome model with little data is extremely risky.

IPW estimators are a general approach to missing-data problems that obey some CI assumptions. While IPW is an old idea in statistics that dates back to Horvitz and Thompson (1952), biostatisticians and econometricians have been actively working on extending it to handle modern problems and estimation methods. See Robins and Rotnitzky (1995); Robins, Rotnitzky, and Zhao (1994, 1995); and Wooldridge (2002, 2007). IPW has been used extensively in the modern treatment-effect estimation literature. See Imbens (2000); Hirano, Imbens, and Ridder (2003); Tan (2010); Wooldridge (2010, chap. 19); van der Laan and Robins (2003); and Tsiatis (2006, chap. 6).

To see the intuition behind IPW, consider a study with observed outcome variable  $y$ , treatment variable  $t \in \{0, 1\}$ , and potential outcomes  $y_0$  and  $y_1$ . As part of this process, we need to estimate

the POM for treatment  $t = 1$ ,  $E(y_1)$ . Using the observed data,  $y_i t_i$  is  $y_{1i}$  when  $t = 1$ , but  $y_{1i}$  is unobserved when  $t = 0$ . An IPW estimator for  $E(y_1)$  is  $1/N \sum_{i=1}^N y_i t_i / p(\mathbf{x}_i)$ , where  $p(\mathbf{x}_i)$  is the probability that  $t_i = 1$  and is a function of the covariates  $\mathbf{x}_i$ . If  $y_{1i}$  were always observed, the weights would all equal 1. This IPW estimator places a larger weight on those observations for which  $y_{1i}$  is observed even though its observation was not likely.

## AIPW estimators

Instead of modeling either the outcome, like RA, or the treatment probability, like IPW, augmented inverse-probability-weighted (AIPW) estimators model both the outcome and the treatment probability. A surprising fact is that only one of the two models must be correctly specified to consistently estimate the treatment effects, a property of the AIPW estimators known as being “doubly robust”. Given that two models instead of one are used, it is less surprising that the AIPW estimators can be more efficient than either the RA or the IPW estimators, though deriving this result is rather technical and relies on the theory of semiparametric estimators.

Intuitively, the AIPW estimator is an IPW that includes an augmentation term that corrects the estimator when the treatment model is misspecified. When the treatment is correctly specified, the augmentation term vanishes as the sample size becomes large. Like the IPW, the AIPW does not perform well when the predicted treatment probabilities are too close to zero or one.

AIPW estimators emerge naturally from a technique of producing more efficient estimators from estimators that have a few main parameters of interest and some auxiliary, or nuisance, parameters used in estimating the few main parameters. This method constructs efficient estimating equations for the main parameters that are orthogonal to the auxiliary parameters. The estimators produced by this method are known as efficient-influence function (EIF) estimators.

To gain some intuition, consider finding an EIF estimator from an IPW estimator for two POMs. Note that we only care about the two POM parameters and not about the many auxiliary parameters used to estimate the treatment probabilities. EIF estimators project the equations that yield the POM parameters onto the equations that yield the auxiliary treatment-model parameters and then use the residuals from this projection to estimate the POM parameters.

We refer to these estimators as “AIPW estimators” instead of “EIF estimators” because the former is commonly used in the biostatistical literature for some of the binary-treatment estimators and because the term “augmented inverse-probability-weighted” tells more about how these estimators relate to the other implemented estimators; see [Tsiatis \(2006\)](#) and [Tan \(2010\)](#). The estimators implemented in `teffects aipw` with the `wls` option are based on those of [Rubin and van der Laan \(2008\)](#), which did well in simulations reported by [Tan \(2010\)](#), and denoted as  $\tilde{\alpha}_{RV}(\hat{\pi})$  in [Tan \(2010, 663\)](#).

When either the outcome model or the treatment model is well specified, the AIPW estimators implemented in `teffects aipw` are more robust than either the RA or the IPW estimators because the AIPW estimators are doubly robust but the RA and IPW estimators are not. When both the outcome and the treatment model are misspecified, which estimator is more robust is a matter of debate in the literature; see [Kang and Schafer \(2007\)](#) and [Robins et al. \(2007\)](#) for some debate, and see [Tan \(2010\)](#) for a more recent discussion.

To the best of our knowledge, there is no general solution to the question of which estimator performs best when both the outcome and the treatment models are misspecified. We suspect that the answer depends on the true models, the implemented specifications, and the polynomials in the covariates used. To help users through this process, the estimators implemented in `teffects` offer many functional forms to approximate either the outcome process or the treatment process. In addition, Stata’s factor-variable notation makes it easy to include polynomials in the covariates. Both of these approximation methods rely on having enough data. `teffects` also makes it easy to compare the results produced by different estimators.



The literature on these methods is vast and growing. For double-robust results and explanations, see [Robins and Rotnitzky \(1995\)](#); [Robins, Rotnitzky, and Zhao \(1995\)](#); [van der Laan and Robins \(2003, chap. 6\)](#); [Bang and Robins \(2005\)](#); [Tsiatis \(2006, chap. 13\)](#); [Wooldridge \(2007; 2010, sec. 21.3.4\)](#); and [Tan \(2010\)](#).

## IPWRA estimators

Like AIPW estimators, inverse-probability-weighted regression-adjustment (IPWRA) estimators combine models for the outcome and treatment status; also like AIPW estimators, IPWRA estimators are doubly robust. IPWRA estimators emerge naturally from a robust approach to missing-data methods. IPWRA estimators use the inverse of the estimated treatment-probability weights to estimate missing-data-corrected regression coefficients that are subsequently used to compute the POMs.

As far as we know, there is no literature that compares the relative efficiency of AIPW estimators, which emerge from a general approach to creating efficient estimators, and the IPWRA estimators, which emerge from a robust-correction approach to missing-data analysis.

The IPWRA estimators are also known as “Wooldridge’s double-robust” estimators because they were derived in [Wooldridge \(2007\)](#) and discussed at length in [Wooldridge \(2010, section 21.3.4\)](#).

## Nearest-neighbor matching estimators

Matching estimators use an average of the outcomes of the nearest individuals to impute the missing potential outcome for each sampled individual. The difference between the observed outcome and the imputed potential outcome is an estimate of the individual-level treatment effect. These estimated individual-level treatment effects are averaged to estimate the ATE or the ATET.

`teffects nnmatch` determines the “nearest” by using a weighted function of the covariates for each observation. This type of matching is known as nearest-neighbor matching (NNM). `teffects psmatch` determines the “nearest” by using the estimated treatment probabilities, which are known as the propensity scores. This second type of matching is known as propensity-score matching (PSM).

NNM is nonparametric in that no explicit functional form for either the outcome model or the treatment model is specified. This flexibility comes at a price; the estimator needs more data to get to the true value than an estimator that imposes a functional form. More formally, the NNM estimator converges to the true value at a rate slower than the parametric rate, which is the square root of the sample size, when matching on more than one continuous covariate. `teffects nnmatch` uses bias correction to fix this problem. PSM provides an alternative to bias correction because it matches on a single continuous covariate, the estimated treatment probabilities.

[Abadie and Imbens \(2006, 2011\)](#) derived the rate of convergence of the NNM estimator and the bias-corrected NNM estimator and the large-sample distributions of the NNM and the bias-corrected NNM estimators. These articles provided the formal results that built on methods suggested in [Rubin \(1973, 1977\)](#).

`teffects nnmatch` is based on the results in [Abadie and Imbens \(2006, 2011\)](#) and a previous implementation in [Abadie et al. \(2004\)](#).



## Propensity-score matching estimators

Instead of performing bias correction to handle the case of more than one continuous covariate, a common solution is to combine all the covariate information into estimated treatment probabilities, known as propensity scores, and use this single continuous covariate as the matching variable.

The term “propensity score” is widely used, but we continue to refer to it as the “treatment probability” to be consistent with the other estimators. We call the estimator that matches on the estimated treatment probabilities the “propensity-score matching (PSM) estimator” because the latter term is ubiquitous.

In effect, the PSM estimator parameterizes the bias-correction term in the treatment probability model. One advantage of matching on the estimated treatment probabilities over the bias-correction method is that one can explore the fit of different treatment probability models using standard methods before performing the nonparametric matching. For example, one can select the treatment model by minimizing an information criterion under i.i.d. sampling. We know of no counterpart for selecting the proper order of the bias-correction term for the NNM estimator.

Matching on estimated treatment probability models has been very popular since [Rosenbaum and Rubin \(1983\)](#) showed that if adjusting for covariates  $\mathbf{x}_i$  is sufficient to estimate the effects, then one can use the probability of treatment to perform the adjustment. [Abadie and Imbens \(2012\)](#) derived a method to estimate the standard errors of the estimator that matches on estimated treatment probabilities, and this method is implemented in `teffects psmatch`.

## Choosing among estimators

There is no definitive way to select one of the estimators implemented in `teffects` over the others. We offer three observations about the tradeoffs among the estimators.

First, if the outcome model is correctly specified, the RA estimator will break down more slowly than the IPW, AIPW, IPWRA, or PSM estimators as the overlap assumption begins to fail. This result depends critically on the ability of the RA estimator to predict into regions in which there are little data.

Second, if the overlap assumption holds, the AIPW and IPWRA estimators have the double-robust property for some functional form combinations. The double-robust property says that if either the outcome model or the treatment model is correctly specified, we can consistently estimate the effects. The properties of double-robust estimators when both models are misspecified are not known, although there is some discussion in the literature about the properties of the AIPW estimators; see [Kang and Schafer \(2007\)](#), [Robins et al. \(2007\)](#), and [Tan \(2010\)](#).

Third, all the estimators require the same assumptions, so if each is correctly specified, they should all produce similar results. Of course, just because they produce similar results does not mean that they are correctly specified; it is possible that they are just behaving similarly in response to some underlying problem.

## Model choice

`teffects` offers a broad selection of functional form combinations so that you can choose a combination that fits your data. Picking a functional form that respects the values of the observed outcomes is usually best. Select `linear` for continuous outcomes over the real line; `logit`, `probit`, or `hetprobit` for binary outcomes; and `poisson` for counts or nonnegative outcomes.

For binary treatments, you can select among `logit`, `probit`, or `hetprobit` models. For multivalued treatments, only the multinomial logit is available to model the treatment probabilities.

Selecting a functional form of a given set of covariates is a famously difficult problem in statistics. In the treatment-effects context, [Cattaneo, Drukker, and Holland \(2013\)](#) found that model selection by minimizing an information criterion worked well. [Cattaneo, Drukker, and Holland \(2013\)](#) discuss a method and a community-contributed command to facilitate the process.

## Acknowledgments

We thank Matias Cattaneo, Guido Imbens, Barbara Sianesi, and Jeffrey Wooldridge for helpful conversations and advice about the literature and methods in this area.

We thank Matias Cattaneo for his collaboration on [Cattaneo, Drukker, and Holland \(2013\)](#) and for many conversations about [Cattaneo \(2010\)](#), a part of which is implemented in `teffects ipw` and `teffects aipw`.

We thank Ashley Holland for her collaboration on [Cattaneo, Drukker, and Holland \(2013\)](#).

We thank Alberto Abadie, Jane Herr, and Guido Imbens for collaboration on [Abadie et al. \(2004\)](#), which is implemented in `teffects nnmatch`.

We thank Matias Cattaneo for sharing an extract of the dataset that he used in [Cattaneo \(2010\)](#) and Barbara Sianesi for sharing an extract of the dataset that she used in [Blundell, Dearden, and Sianesi \(2005\)](#).

We thank Alberto Abadie, Guido Imbens, and László Sándor for sharing their implementation of the standard errors derived in [Abadie and Imbens \(2012\)](#), which are implemented in `teffects psmatch`.

We thank Edwin Leuven and Barbara Sianesi for their inspirational command `psmatch2`, which computes several of the estimators in `teffects` and a few more that we have not implemented.

## References

- Abadie, A., D. M. Drukker, J. L. Herr, and G. W. Imbens. 2004. [Implementing matching estimators for average treatment effects in Stata](#). *Stata Journal* 4: 290–311.
- Abadie, A., and G. W. Imbens. 2006. Large sample properties of matching estimators for average treatment effects. *Econometrica* 74: 235–267. <https://doi.org/10.1111/j.1468-0262.2006.00655.x>.
- . 2011. Bias-corrected matching estimators for average treatment effects. *Journal of Business and Economic Statistics* 29: 1–11. <https://doi.org/10.1198/jbes.2009.07333>.
- . 2012. Matching on the estimated propensity score. Harvard University and National Bureau of Economic Research. <http://www.nber.org/papers/w15301>.
- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.
- Bang, H., and J. M. Robins. 2005. Doubly robust estimation in missing data and causal inference models. *Biometrics* 61: 962–972. <https://doi.org/10.1111/j.1541-0420.2005.00377.x>.
- Becker, S. O., and M. Caliendo. 2007. Sensitivity analysis for average treatment effects. *Stata Journal* 7: 71–83.
- Blundell, R. W., L. Dearden, and B. Sianesi. 2005. Evaluating the effect of education on earnings: Models, methods and results from the National Child Development Survey. *Journal of the Royal Statistical Society, Series A* 168: 473–512. <https://doi.org/10.1111/j.1467-985X.2004.00360.x>.
- Brown, G. K., and T. Mergoupis. 2011. [Treatment interactions with nonexperimental data in Stata](#). *Stata Journal* 11: 545–555.
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.

- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.
- Cattaneo, M. D., R. Titiunik, and G. Vazquez-Bare. 2020. Analysis of regression-discontinuity designs with multiple cutoffs or multiple scores. *Stata Journal* 20: 866–891.
- Cerulli, G. 2015a. *Econometric Evaluation of Socio-Economic Programs: Theory and Applications*. Berlin: Springer.
- . 2015b. `ctreatreg`: Command for fitting dose-response models under exogenous and endogenous treatment. *Stata Journal* 15: 1019–1045.
- . 2017. Identification and estimation of treatment effects in the presence of (correlated) neighborhood interactions: Model and Stata implementation via `ntreatreg`. *Stata Journal* 17: 803–833.
- Daniel, R. M., B. L. De Stavola, and S. N. Cousens. 2011. `gformula`: Estimating causal effects in the presence of time-varying confounding or mediation using the `g`-computation formula. *Stata Journal* 11: 479–517.
- Deb, P., E. C. Norton, and W. G. Manning. 2017. *Health Econometrics Using Stata*. College Station, TX: Stata Press.
- Drukker, D. M. 2016. A generalized regression-adjustment estimator for average treatment effects from panel data. *Stata Journal* 16: 826–836.
- Galiani, S., and B. Quistorff. 2017. The `synth_runner` package: Utilities to automate synthetic control estimation using `synth`. *Stata Journal* 17: 834–849.
- Heckman, J. 1997. Instrumental variables: A study of implicit behavioral assumptions used in making program evaluations. *Journal of Human Resources* 32: 441–462. <https://doi.org/10.2307/146178>.
- Heckman, J., and S. Navarro-Lozano. 2004. Using matching, instrumental variables, and control functions to estimate economic choice models. *Review of Economics and Statistics* 86: 30–57. <https://doi.org/10.1162/003465304323023660>.
- Heß, S. 2017. Randomization inference with Stata: A guide and software. *Stata Journal* 17: 630–651.
- Hirano, K., G. W. Imbens, and G. Ridder. 2003. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica* 71: 1161–1189. <https://doi.org/10.1111/1468-0262.00442>.
- Holland, P. W. 1986. Statistics and causal inference. *Journal of the American Statistical Association* 81: 945–960. <https://doi.org/10.2307/2289064>.
- Horvitz, D. G., and D. J. Thompson. 1952. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association* 47: 663–685. <https://doi.org/10.1080/01621459.1952.10483446>.
- Imbens, G. W. 2000. The role of the propensity score in estimating dose–response functions. *Biometrika* 87: 706–710. <https://doi.org/10.1093/biomet/87.3.706>.
- . 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics* 86: 4–29. <https://doi.org/10.1162/003465304323023651>.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.
- Kang, J. D. Y., and J. L. Schafer. 2007. Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data. *Statistical Science* 22: 523–539. <https://doi.org/10.1214/07-STS227>.
- Lane, P. W., and J. A. Nelder. 1982. Analysis of covariance and standardization as instances of prediction. *Biometrics* 38: 613–621. <https://doi.org/10.2307/2530043>.
- Nichols, A. 2007. Causal inference with observational data. *Stata Journal* 7: 507–541.
- Robins, J. M. 1986. A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical Modeling* 7: 1393–1512. [https://doi.org/10.1016/0270-0255\(86\)90088-6](https://doi.org/10.1016/0270-0255(86)90088-6).
- Robins, J. M., and A. Rotnitzky. 1995. Semiparametric efficiency in multivariate regression models with missing data. *Journal of the American Statistical Association* 90: 122–129. <https://doi.org/10.2307/2291135>.
- Robins, J. M., A. Rotnitzky, and L. P. Zhao. 1994. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association* 89: 846–866. <https://doi.org/10.2307/2290910>.
- . 1995. Analysis of semiparametric regression models for repeated outcomes in the presence of missing data. *Journal of the American Statistical Association* 90: 106–121. <https://doi.org/10.1080/01621459.1995.10476493>.

- Robins, J. M., M. Sued, Q. Lei-Gomez, and A. Rotnitzky. 2007. Comment: Performance of double-robust estimators when “inverse probability” weights are highly variable. *Statistical Science* 22: 544–559. <https://doi.org/10.1214/07-STS227D>.
- Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55. <https://doi.org/10.2307/2335942>.
- Rubin, D. B. 1973. Matching to remove bias in observational studies. *Biometrics* 29: 159–183. <https://doi.org/10.2307/2529684>.
- . 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701. <https://doi.org/10.1037/h0037350>.
- . 1977. Assignment to treatment group on the basis of a covariate. *Journal of Educational Statistics* 2: 1–26. <https://doi.org/10.2307/1164933>.
- Rubin, D. B., and M. J. van der Laan. 2008. Empirical efficiency maximization: Improved locally efficient covariate adjustment in randomized experiments and survival analysis. *International Journal of Biostatistics* 4: 1–40. <https://doi.org/10.2202/1557-4679.1084>.
- Sasaki, Y., and T. Ura. 2022. Average treatment effect estimates robust to the “limited overlap” problem: robustate. *Stata Journal* 22: 344–354.
- Tan, Z. 2010. Bounded, efficient and doubly robust estimation with inverse weighting. *Biometrika* 97: 661–682. <https://doi.org/10.1093/biomet/asq035>.
- Tsiatis, A. A. 2006. *Semiparametric Theory and Missing Data*. New York: Springer.
- van der Laan, M. J., and J. M. Robins. 2003. *Unified Methods for Censored Longitudinal Data and Causality*. New York: Springer.
- Vittinghoff, E., D. V. Glidden, S. C. Shiboski, and C. E. McCulloch. 2012. *Regression Methods in Biostatistics: Linear, Logistic, Survival, and Repeated Measures Models*. 2nd ed. New York: Springer.
- Wooldridge, J. M. 2002. Inverse probability weighted M-estimators for sample selection, attrition, and stratification. *Portuguese Economic Journal* 1: 117–139. <https://doi.org/10.1007/s10258-002-0008-x>.
- . 2007. Inverse probability weighted estimation for general missing data problems. *Journal of Econometrics* 141: 1281–1301. <https://doi.org/10.1016/j.jeconom.2007.02.002>.
- . 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

## Also see

[TE] **teffects** — Treatment-effects estimation for observational data

[TE] **teffects intro** — Introduction to treatment effects for observational data

[TE] **teffects multivalued** — Multivalued treatment effects

# Title

**teffects aipw** — Augmented inverse-probability weighting

<a href="#">Description</a>	<a href="#">Quick start</a>	<a href="#">Menu</a>	<a href="#">Syntax</a>
<a href="#">Options</a>	<a href="#">Remarks and examples</a>	<a href="#">Stored results</a>	<a href="#">Methods and formulas</a>
<a href="#">References</a>	<a href="#">Also see</a>		

## Description

`teffects aipw` estimates the average treatment effect (ATE) and the potential-outcome means (POMs) from observational data by augmented inverse-probability weighting (AIPW). AIPW estimators combine aspects of regression-adjustment and inverse-probability-weighted methods. AIPW estimators have the double-robust property. `teffects aipw` accepts a continuous, binary, count, fractional, or nonnegative outcome and allows a multivalued treatment.

See [\[TE\] teffects intro](#) or [\[TE\] teffects intro advanced](#) for more information about estimating treatment effects from observational data.

## Quick start

ATE of binary treatment `treat2` by AIPW using a linear model for outcome `y1` on `x1` and `x2` and a logistic model for `treat2` on `x1` and `w`

```
teffects aipw (y1 x1 x2) (treat2 x1 w)
```

As above, but use a fractional logistic model for fractional outcome `y2`

```
teffects aipw (y2 x1 x2, flogit) (treat2 x1 w)
```

As above, but use a heteroskedastic probit model for binary outcome `y3` and a probit model for `treat2`

```
teffects aipw (y3 x1 x2, hetprobit(x1 x2)) (treat2 x1 w, probit)
```

ATE for each level of three-valued treatment `treat3` on `y1`

```
teffects aipw (y1 x1 x2) (treat3 x1 w)
```

As above, and specify that `treat3 = 3` is the control level

```
teffects aipw (y1 x1 x2) (treat3 x1 w), control(3)
```

Same as above, specified using the label “MyControl” corresponding to `treat3 = 3`

```
teffects aipw (y1 x1 x2) (treat3 x1 w), control(MyControl)
```

## Menu

Statistics > Treatment effects > Continuous outcomes > Augmented inverse-probability weighting

Statistics > Treatment effects > Binary outcomes > Augmented inverse-probability weighting

Statistics > Treatment effects > Count outcomes > Augmented inverse-probability weighting

Statistics > Treatment effects > Fractional outcomes > Augmented inverse-probability weighting

Statistics > Treatment effects > Nonnegative outcomes > Augmented inverse-probability weighting

## Syntax

```
teffects aipw (ovar omvarlist [, omodel noconstant])
              (tvar tmvarlist [, tmodel noconstant]) [if] [in] [weight]
              [, stat options]
```

*ovar* is a binary, count, continuous, fractional, or nonnegative outcome of interest.

*omvarlist* specifies the covariates in the outcome model.

*tvar* must contain integer values representing the treatment levels.

*tmvarlist* specifies the covariates in the treatment-assignment model.

<i>omodel</i>	Description
---------------	-------------

Model

<code>linear</code>	linear outcome model; the default
<code>logit</code>	logistic outcome model
<code>probit</code>	probit outcome model
<code>hetprobit(<i>varlist</i>)</code>	heteroskedastic probit outcome model
<code>poisson</code>	exponential outcome model
<code>flogit</code>	fractional logistic outcome model
<code>fprobit</code>	fractional probit outcome model
<code>fheterprobit(<i>varlist</i>)</code>	fractional heteroskedastic probit outcome model

*omodel* specifies the model for the outcome variable.

<i>tmodel</i>	Description
---------------	-------------

Model

<code>logit</code>	logistic treatment model; the default
<code>probit</code>	probit treatment model
<code>hetprobit(<i>varlist</i>)</code>	heteroskedastic probit treatment model

*tmodel* specifies the model for the treatment variable.

For multivalued treatments, only `logit` is available and multinomial logit is used.

<i>stat</i>	Description
-------------	-------------

Stat

<code>ate</code>	estimate average treatment effect in population; the default
<code>pomeans</code>	estimate potential-outcome means

<i>options</i>	Description
Model	
<code>nls</code>	estimate conditional means by nonlinear least squares
<code>wnls</code>	estimate conditional means by weighted nonlinear least squares
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>robust</code> , <code>cluster <i>clustvar</i></code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>aequations</code>	display auxiliary-equation results
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
Advanced	
<code>pstolerance(#)</code>	set tolerance for overlap assumption
<code>osample(<i>newvar</i>)</code>	<i>newvar</i> identifies observations that violate the overlap assumption
<code>control(#   <i>label</i>)</code>	specify the level of <i>tvar</i> that is the control
<code>coeflegend</code>	display legend instead of statistics

*omvarlist* and *tmvarlist* may contain factor variables; see [U] 11.4.3 Factor variables.

`bootstrap`, `by`, `collect`, `jackknife`, and `statsby` are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the `bootstrap` prefix; see [R] `bootstrap`.

`fweights` and `iweights` are allowed; see [U] 11.1.6 weight.

`coeflegend` does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

## Options

### Model

`noconstant`; see [R] Estimation options.

`nls` specifies that the parameters of the outcome model be estimated by nonlinear least squares instead of the default maximum likelihood.

`wnls` specifies that the parameters of the outcome model be estimated by weighted nonlinear least squares instead of the default maximum likelihood. The weights make the estimator of the effect parameters more robust to a misspecified outcome model.

### Stat

*stat* is one of two statistics: `ate` or `pomeans`. `ate` is the default.

`ate` specifies that the average treatment effect be estimated.

`pomeans` specifies that the potential-outcome means for each treatment level be estimated.

## SE/Robust

`vce(vctype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster clustvar`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce\\_option](#).

## Reporting

`level(#)`; see [R] [Estimation options](#).

`aequations` specifies that the results for the outcome-model or the treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

`display_options`: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

## Maximization

`maximize_options`: `iterate(#)`, `[no]log`, and `from(init_specs)`; see [R] [Maximize](#). These options are seldom used.

`init_specs` is one of

`matname [ , skip copy ]`

`# [ , # ... ] , copy`

## Advanced

`pstolerance(#)` specifies the tolerance used to check the overlap assumption. The default value is `pstolerance(1e-5)`. `teffects` will exit with an error if an observation has an estimated propensity score smaller than that specified by `pstolerance()`.

`osample(newvar)` specifies that indicator variable `newvar` be created to identify observations that violate the overlap assumption.

`control(# | label)` specifies the level of `tvar` that is the control. The default is the first treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with statistic `pomeans`.

The following option is available with `teffects aipw` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

## Remarks and examples

Remarks are presented under the following headings:

[Overview](#)

[Video example](#)

## Overview

AIPW estimators use inverse-probability weights to correct for the missing-data problem arising from the fact that each subject is observed in only one of the potential outcomes; these estimators also use an augmentation term in the outcome model to correct the estimator in case the treatment model is misspecified. If the treatment model is correctly specified, the augmentation term goes to zero in large samples.



AIPW estimators compute averages of the augmented inverse-probability-weighted outcomes for each treatment level. Contrasts of these averages provide estimates of the treatment effects.

AIPW estimators use a model to predict treatment status, and they use another model to predict outcomes. Because of the double-robust property, only one of these two models must be correctly specified for the AIPW estimator to be consistent.

AIPW estimators use a three-step approach to estimating treatment effects:

1. They estimate the parameters of the treatment model and compute inverse-probability weights.
2. They estimate separate regression models of the outcome for each treatment level and obtain the treatment-specific predicted outcomes for each subject.
3. They compute the weighted means of the treatment-specific predicted outcomes, where the weights are the inverse-probability weights computed in step 1. The contrasts of these weighted averages provide the estimates of the ATEs.

These steps produce consistent estimates of the effect parameters because the treatment is assumed to be independent of the potential outcomes after conditioning on the covariates. The overlap assumption ensures that predicted inverse-probability weights do not get too large. The standard errors reported by `teffects aipw` correct for the three-step process. See [TE] [teffects intro](#) or [TE] [teffects intro advanced](#) for more information about this estimator.

We will illustrate the use of `teffects aipw` by using data from a study of the effect of a mother's smoking status during pregnancy (`mbsmoke`) on infant birthweight (`bweight`) as reported by Cattaneo (2010). This dataset also contains information about each mother's age (`mage`), education level (`medu`), marital status (`mmarried`), whether the first prenatal exam occurred in the first trimester (`prenatal1`), and whether this baby was the mother's first birth (`fbaby`).

### ▷ Example 1: Estimating the ATE

We begin by using `teffects aipw` to estimate the average treatment effect of `mbsmoke` on `bweight`. We use a probit model to predict treatment status as a function of `mmarried`, `mage`, and `fbaby`; to maximize the predictive power of this model, we use factor-variable notation to incorporate quadratic effects of the mother's age, the only continuous covariate in our model. We use linear regression to model birthweight, using `prenatal1`, `mmarried`, `mage`, and `fbaby` as explanatory variables. We type

```

. use https://www.stata-press.com/data/r17/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects aipw (bweight prenatal1 mmarried mage fbaby)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, probit)

Iteration 0:   EE criterion = 4.629e-21
Iteration 1:   EE criterion = 1.944e-25

Treatment-effects estimation           Number of obs   =       4,642
Estimator      : augmented IPW
Outcome model  : linear by ML
Treatment model: probit

```

bweight	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
mbsmoke (Smoker vs Nonsmoker)	-230.9892	26.21056	-8.81	0.000	-282.361	-179.6174
POmean						
mbsmoke Nonsmoker	3403.355	9.568472	355.68	0.000	3384.601	3422.109

The average birthweight if all mothers were to smoke would be 231 grams less than the average of 3,403 grams that would occur if none of the mothers had smoked.

◀

By default, `teffects aipw` reports the ATE and the POM for the base (untreated) subjects. The `pomeans` option allows us to view the treated subjects' POM as well; the `aequations` option displays the regression model coefficients used to predict the POMs as well as the coefficients from the model used to predict treatment.

### ▶ Example 2: Displaying the POMs and equations

Here we use the `pomeans` and `aequations` options to obtain estimates of both POMs and view all the fitted equations underlying our estimates:

```
. teffects aipw (bweight prenatal1 mmarried mage fbaby)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, probit), pomeans aequations
Iteration 0: EE criterion = 4.629e-21
Iteration 1: EE criterion = 6.856e-26
Treatment-effects estimation                Number of obs    =    4,642
Estimator      : augmented IPW
Outcome model  : linear by ML
Treatment model: probit
```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
<b>POmeans</b>						
mbsmoke						
Nonsmoker	3403.355	9.568472	355.68	0.000	3384.601	3422.109
Smoker	3172.366	24.42456	129.88	0.000	3124.495	3220.237
<b>OME0</b>						
prenatal1	64.40859	27.52699	2.34	0.019	10.45669	118.3605
mmarried	160.9513	26.6162	6.05	0.000	108.7845	213.1181
mage	2.546828	2.084324	1.22	0.222	-1.538373	6.632028
fbaby	-71.3286	19.64701	-3.63	0.000	-109.836	-32.82117
_cons	3202.746	54.01082	59.30	0.000	3096.886	3308.605
<b>OME1</b>						
prenatal1	25.11133	40.37541	0.62	0.534	-54.02302	104.2457
mmarried	133.6617	40.86443	3.27	0.001	53.5689	213.7545
mage	-7.370881	4.21817	-1.75	0.081	-15.63834	.8965804
fbaby	41.43991	39.70712	1.04	0.297	-36.38461	119.2644
_cons	3227.169	104.4059	30.91	0.000	3022.537	3431.801
<b>TME1</b>						
mmarried	-.6484821	.0554173	-11.70	0.000	-.757098	-.5398663
mage	.1744327	.0363718	4.80	0.000	.1031452	.2457202
c.mage#						
c.mage	-.0032559	.0006678	-4.88	0.000	-.0045647	-.0019471
fbaby	-.2175962	.0495604	-4.39	0.000	-.3147328	-.1204595
medu	-.0863631	.0100148	-8.62	0.000	-.1059917	-.0667345
_cons	-1.558255	.4639691	-3.36	0.001	-2.467618	-.6488926

The coefficient table indicates that the treated POM is 3,172 grams, 231 grams less than the untreated POM. The sections of the table labeled OME0 and OME1 represent the linear regression coefficients for the untreated and treated potential-outcome equations, respectively. The coefficients of the TME1 equation are used in the probit model to predict treatment status.

◀

As is well known, the standard probit model assumes that the error terms in the latent-utility framework are homoskedastic; the model is not robust to departures from this assumption. An alternative is to use the heteroskedastic probit model, which explicitly models the error variance as a function of a set of variables.

### ► Example 3: Heteroskedastic probit treatment model

Here we refit our model as in the previous examples, but we instead use heteroskedastic probit to model the treatment variable. We posit that the heteroskedasticity is a function of the mother's age. We type

```

. teffects aipw (bweight prenatal1 mmarried fbaby)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, hetprobit(c.mage)), aequations
Iteration 0: EE criterion = 1.746e-19
Iteration 1: EE criterion = 1.746e-19 (backed up)
Treatment-effects estimation          Number of obs    =      4,642
Estimator      : augmented IPW
Outcome model  : linear by ML
Treatment model: heteroskedastic probit

```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
<b>ATE</b>						
mbsmoke (Smoker vs Nonsmoker)	-230.2699	27.49461	-8.38	0.000	-284.1584	-176.3815
<b>POmean</b>						
mbsmoke Nonsmoker	3403.657	9.540713	356.75	0.000	3384.957	3422.356
<b>OME0</b>						
prenatal1	69.5048	27.04642	2.57	0.010	16.49479	122.5148
mmarried	173.74	24.63865	7.05	0.000	125.4491	222.0308
fbaby	-79.19473	18.62584	-4.25	0.000	-115.7007	-42.68875
_cons	3260.768	28.29282	115.25	0.000	3205.315	3316.221
<b>OME1</b>						
prenatal1	12.86437	39.83916	0.32	0.747	-65.21894	90.94768
mmarried	113.3491	39.47422	2.87	0.004	35.9811	190.7172
fbaby	64.22326	38.42042	1.67	0.095	-11.07939	139.5259
_cons	3051.268	37.30413	81.79	0.000	2978.153	3124.383
<b>TME1</b>						
mmarried	-.3551755	.1044199	-3.40	0.001	-.5598347	-.1505162
mage	.0831898	.0349088	2.38	0.017	.0147699	.1516097
c.mage# c.mage	-.0013458	.0006659	-2.02	0.043	-.002651	-.0000406
fbaby	-.1170697	.044998	-2.60	0.009	-.2052643	-.0288752
medu	-.0435057	.0147852	-2.94	0.003	-.0724842	-.0145272
_cons	-.8757021	.347814	-2.52	0.012	-1.557405	-.1939993
<b>TME1_lnsigma</b>						
mage	-.0236336	.0107134	-2.21	0.027	-.0446315	-.0026357

The equation labeled TME1\_lnsigma represents the heteroskedasticity function used to model the logarithm of the variance. Because the coefficient on the single variable we specified is significant below the 5% level, we conclude that allowing for heteroskedasticity was indeed necessary.

◀

Rather than using maximum likelihood to fit the outcome model, you can instruct **teffects aipw** to use a weighted nonlinear least-squares (WNLS) estimator instead. The WNLS estimator may be more robust to outcome model misspecification.

## ► Example 4: Using the WNLS estimator

Here we use WNLS to fit our outcome model:

```
. use https://www.stata-press.com/data/r17/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects aipw (bweight prenatal1 mmarried mage fbaby)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, probit), wnls

Iteration 0:  EE criterion = 2.742e-20
Iteration 1:  EE criterion = 3.436e-24

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : augmented IPW
Outcome model  : linear by WNLS
Treatment model: probit
```

		Robust				
	bweight	Coefficient	std. err.	z	P> z	[95% conf. interval]
ATE						
	mbsmoke (Smoker vs Nonsmoker)	-227.1956	27.34794	-8.31	0.000	-280.7966 -173.5946
POmean						
	mbsmoke Nonsmoker	3403.251	9.596622	354.63	0.000	3384.442 3422.06

The ATE of  $-227$  is slightly greater than the ATE of  $-231$  estimated in [example 1](#). The estimated POMs are nearly indistinguishable.

◀

## Video example

[Treatment effects: Augmented inverse-probability weighting](#)

## Stored results

`teffects aipw` stores the following in `e()`:

### Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level $j$
<code>e(N_clust)</code>	number of clusters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(converged)</code>	1 if converged, 0 otherwise

### Macros

<code>e(cmd)</code>	<code>teffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(subcmd)</code>	<code>aipw</code>
<code>e(tmodel)</code>	<code>logit</code> , <code>probit</code> , or <code>hetprobit</code>
<code>e(omodel)</code>	<code>linear</code> , <code>logit</code> , <code>probit</code> , <code>hetprobit</code> , <code>poisson</code> , <code>flogit</code> , <code>fprobit</code> , or <code>fhnetprobit</code>
<code>e(stat)</code>	statistic estimated, <code>ate</code> or <code>pomeans</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(cme)</code>	<code>ml</code> , <code>nls</code> , or <code>wls</code>
<code>e(vce)</code>	<i>vctype</i> specified in <code>vce()</code>
<code>e(vctype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

### Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

### Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

### Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, $p$ -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r`-class command is run after the estimation command.

## Methods and formulas

The methods and formulas presented here provide the technical details underlying the estimators implemented in `teffects ra`, `teffects ipw`, `teffects aipw`, and `teffects ipwra`. See [Methods and formulas](#) of [TE] `teffects nnmatch` for the methods and formulas used by `teffects nnmatch` and `teffects psmatch`.

Methods and formulas are presented under the following headings:

- Parameters and notation*
- Overview of EE estimators*
- VCE for EE estimators*
- TM and OM estimating functions*
  - TM estimating functions*
  - OM estimating functions*
- Effect estimating functions*
  - RA estimators*
  - IPW estimators*
  - AIPW estimators*
  - IPWRA estimators*

## Parameters and notation

We begin by reviewing the effect parameters estimated by `teffects` and some essential notation.

The potential outcome that an individual would obtain if given treatment level  $t \in \{0, 1, \dots, q\}$  is  $y_t$ . Each  $y_t$  is a random variable, the realizations of which are  $y_{ti}$ . Throughout this document,  $i$  subscripts denote realizations of the corresponding, unsubscripted random variables.

The three parameters of interest are

1. the potential-outcome mean (POM)  $\alpha_t = E(y_t)$ ;
2. the average treatment effect (ATE)  $\tau_t = E(y_t - y_0)$ ; and
3. the average treatment effect on the treated (ATET)  $\delta_t = E(y_t - y_0 | t = \tilde{t})$ .

The no-treatment level is 0.

The estimators implemented in `teffects` use three assumptions to justify the equations used for estimation and inference about the effect parameters of interest:

1. Conditional mean independence (CMI) allows us to estimate potential-outcome means from the observed outcomes in the sample.
2. Overlap ensures that we have data on each type of individual in each treatment level.
3. Independent observations ensure that the outcome and treatment for one individual has no effect on the outcome or treatment for any other individual.

`teffects ra` implements some regression-adjustment (RA) estimators; `teffects ipw` implements some inverse-probability-weighted (IPW) estimators; `teffects ipwra` implements some inverse-probability-weighted regression-adjustment (IPWRA) estimators; and `teffects aipw` implements some augmented inverse-probability-weighted (AIPW) estimators. All are implemented as estimating-equation (EE) estimators. The estimators are consistent and asymptotically normally distributed under the CMI, overlap, and independence assumptions.

## Overview of EE estimators

EE estimators compute estimates by solving sample estimating equations. The sample estimating equations are the sample equivalents of population expectation equations.

Each EE estimator specifies a set of estimating equations for the effect parameters of interest and a set of estimating equations for the auxiliary parameters in the outcome model (OM) or the treatment model (TM). The next few sections provide tremendous detail about the estimating equations that define the RA, IPW, AIPW, and IPWRA estimators.

Ignoring the details for a moment, EE estimators solve systems of equations to compute estimates. A standard robust estimator is consistent for the variance of the estimator (VCE). All the details involve the equations specified by choices of estimator and functional forms for the OM or TM.

When used, the OM is a model for the conditional mean of the outcome variable. We let  $\mu(\mathbf{x}, t, \beta_t)$  denote a conditional-mean model for the outcome  $y$  conditional on covariates  $\mathbf{x}$  and treatment level  $t$ . Mathematically,  $E(y|\mathbf{x}, t) = \mu(\mathbf{x}, t, \beta_t)$ , where  $\beta_t$  are the parameters of the conditional-mean model given treatment level  $t$ . The table below provides details about the available functional forms.

Outcome model	Functional form for $\mu(\mathbf{x}, t, \beta_t)$
linear	$\mathbf{x}\beta_t$
logit, flogit	$\exp(\mathbf{x}\beta_t)/\{1 + \exp(\mathbf{x}\beta_t)\}$
probit, fprobit	$\Phi(\mathbf{x}\beta_t)$
poisson	$\exp(\mathbf{x}\beta_t)$
hetprobit, fheterprobit	$\Phi\{\dot{\mathbf{x}}\dot{\beta}_t / \exp(\ddot{\mathbf{x}}\ddot{\beta}_t)\}$

In the cases of `hetprobit` and `fheterprobit`, we use  $\dot{\mathbf{x}}$  and  $\dot{\beta}_t$  to denote the variables and parameters in the index function, and we use  $\ddot{\mathbf{x}}$  and  $\ddot{\beta}_t$  to denote the variables and parameters in the variance equation. We define  $\mathbf{x}' = (\dot{\mathbf{x}}', \ddot{\mathbf{x}}')$  and  $\beta'_t = (\dot{\beta}'_t, \ddot{\beta}'_t)$ .

We write the vector of parameters for the outcome model over all treatment levels as  $\beta' = (\beta'_0, \beta'_1, \dots, \beta'_q)$ .

Next we provide details about the estimating equations implied by each functional form choice.

When used, the TM is a model for the conditional probability of treatment. We let  $p(\mathbf{z}, t, \gamma)$  denote the conditional probability model for the probability that a person receives treatment  $t$ , conditional on covariates  $\mathbf{z}$ . The table below provides details about the functional form options allowed in the case of a binary treatment.

Treatment model	Functional form for $p(\mathbf{z}, t, \gamma)$
logit	$\exp(\mathbf{z}\gamma)/\{1 + \exp(\mathbf{z}\gamma)\}$
probit	$\Phi(\mathbf{z}\gamma)$
hetprobit	$\Phi\{\dot{\mathbf{z}}\dot{\gamma} / \exp(\ddot{\mathbf{z}}\ddot{\gamma})\}$

In the case of `hetprobit`, we use  $\dot{\mathbf{z}}$  and  $\dot{\gamma}$  to denote the variables and parameters in the index function, and we use  $\ddot{\mathbf{z}}$  and  $\ddot{\gamma}$  to represent the variables and parameters in the variance equation. We define  $\mathbf{z}' = (\dot{\mathbf{z}}', \ddot{\mathbf{z}}')$ , and  $\gamma' = (\dot{\gamma}', \ddot{\gamma}')$ .

In the multivalued-treatment case,  $p(\mathbf{z}, t, \gamma)$  is specified as a multinomial logit with  $p(\mathbf{z}, t, \gamma) = \exp(\mathbf{z}\gamma_t) / \{1 + \sum_{k=1}^q \exp(\mathbf{z}\gamma_k)\}$  and  $\gamma' = (\gamma'_1, \gamma'_2, \dots, \gamma'_q)$ . (We present formulas for the case with treatment level 0 as the base with  $\gamma'_0 = \mathbf{0}'$ ; see [R] **mlogit** for background.) In `teffects`, the `logit` option in the treatment-model specification means binary logit for the binary-treatment case and multinomial logit for the multivalued-treatment case: this simplifies the use of the command and makes statistical sense.

Below we provide details about the estimating equations implied by each functional form choice. The effect parameters of interest are

1. the POMs denoted by  $\alpha' = (\alpha_0, \alpha_1, \dots, \alpha_q)$ ;
2. the ATEs denoted by  $\tau' = (\tau_1, \tau_2, \dots, \tau_q)$ ; and
3. the ATETs denoted by  $\delta' = (\delta_1, \delta_2, \dots, \delta_q)$ .



We denote the effect parameters by  $\vartheta$  and all the parameters in any particular case by  $\theta$ . More formally,  $\theta$  is the concatenation of the effect parameters, the OM parameters, and the TM parameters;  $\theta' = (\vartheta', \beta', \gamma')$ , where  $\vartheta$  is  $\alpha$ ,  $\tau$ , or  $\delta$ , and  $\beta$  or  $\gamma$  may not be present, depending on the case at hand.

In the subsections below, we discuss estimators for the elements in  $\theta$  in detail and note how these elements change over the cases defined by effect parameters and estimators. The parameter vector  $\theta$  denotes all the parameters, no matter which particular case is under consideration.

The EE estimators described in this section are defined by a set of equations,

$$E\{\mathbf{s}(\mathbf{x}, \mathbf{z}, \theta)\} = \mathbf{0}$$

where  $\mathbf{s}(\mathbf{x}, \mathbf{z}, \theta)$  is a vector of estimating functions. Note the notation: estimating equations equate the expected value of a vector of estimating functions to zero.

Because each of the estimating functions has mean zero, we can construct estimators that find the estimates  $\hat{\theta}$  by solving a system of equations,

$$1/N \sum_i^N \mathbf{s}_i(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta}) = \mathbf{0}$$

where  $\mathbf{s}_i(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta})$  are the sample realizations of the estimating functions. In other words, the parameter estimates set the average of the realizations of each estimating function to zero. Almost all the details below involve specifying the sample realizations  $\mathbf{s}_i(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta})$ .

Estimators that set the expected value of estimating functions to zero are known as estimating-equations (EE) estimators, M estimators, or Z estimators in the statistics literature and as generalized method of moments (GMM) estimators in the econometrics literature. See [van der Vaart \(1998, 41\)](#), [Stefanski and Boos \(2002\)](#), and [Tsiatis \(2006, sec. 3.2\)](#) for statistics; and see [Wooldridge \(2010, chap. 14\)](#), [Cameron and Trivedi \(2005, chap. 6\)](#), and [Newey and McFadden \(1994\)](#) for econometrics.

We refer to them as EE estimators because this name is closest to being independent of any discipline. The estimators are implemented using `gmm` because they are exactly identified generalized method of moments (GMM) estimators. When weights are specified by the user, they are applied to the estimating equations just as `gmm` applies user-specified weights.

Each estimator has a set of estimating equations for the effect parameters and either an OM or a TM, or both. The OM parameters or the TM parameters are auxiliary parameters used to estimate the effect parameters of interest.

Each set of parameters has its own set of sample estimating equations:

$1/N \sum_i \mathbf{s}_{e,i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta}) = \mathbf{0}$  are the sample estimating equations for the effect parameters. These equations determine the effect parameter estimates  $\hat{\vartheta}$  as functions of the data and the other estimated parameters.

$1/N \sum_i \mathbf{s}_{om,i}(\mathbf{x}_i, w_i, \hat{\beta}) = \mathbf{0}$  are the sample estimating equations for OM parameters that use the weights  $w_i$ , which are functions of the TM parameters.

$1/N \sum_i \mathbf{s}_{tm,i}(\mathbf{z}_i, \hat{\gamma}) = \mathbf{0}$  are the sample estimating equations for TM parameters.

The whole set of sample estimating functions is  $\mathbf{s}_i(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta})$  with

$$\mathbf{s}_i(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta})' = (\mathbf{s}_{e,i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta})', \mathbf{s}_{om,i}(\mathbf{x}_i, w_i(t), \hat{\beta})', \mathbf{s}_{tm,i}(\mathbf{z}_i, \hat{\gamma})')$$

although not all the estimators have each of three components.

## VCE for EE estimators

The Huber/White/robust sandwich estimator is consistent for the variance–covariance of the estimator (VCE). See [van der Vaart \(1998, 41\)](#), [Stefanski and Boos \(2002\)](#), and [Tsiatis \(2006, sec. 3.2\)](#) for statistics; and see [Wooldridge \(2010, chap. 14\)](#), [Cameron and Trivedi \(2005, chap. 6\)](#), and [Newey and McFadden \(1994\)](#) for econometrics.

The formula is

$$\widehat{\mathbf{V}} = (1/N)\overline{\mathbf{G}} \overline{\mathbf{S}} \overline{\mathbf{G}}'$$

where

$$\overline{\mathbf{G}} = \left\{ (1/N) \sum_i \frac{\partial s_i(\mathbf{x}_i, \mathbf{z}_i, \widehat{\boldsymbol{\theta}})}{\partial \widehat{\boldsymbol{\theta}}} \right\}^{-1}$$

and

$$\overline{\mathbf{S}} = (1/N) \sum_i s_i(\mathbf{x}_i, \mathbf{z}_i, \widehat{\boldsymbol{\theta}}) s_i(\mathbf{x}_i, \mathbf{z}_i, \widehat{\boldsymbol{\theta}})'$$

The matrix  $\overline{\mathbf{G}}$  is not symmetric because our EE estimators come from stacking moment conditions instead of optimizing a single objective function. The implication is that the robust formula should always be used because, even under correct specification, the nonsymmetric  $\overline{\mathbf{G}}$  and the symmetric  $\overline{\mathbf{S}}$  converge to different matrices.

## TM and OM estimating functions

Although the sample estimating functions for the effect parameters, the  $s_{e,i}(\mathbf{x}_i, \mathbf{z}_i, \widehat{\boldsymbol{\theta}})$ , are estimator specific, the sample estimating functions for the TM parameters, the  $s_{\text{tm},i}(\mathbf{z}_i, \widehat{\boldsymbol{\gamma}})$ , and the sample estimating functions for the OM parameters, the  $s_{\text{om},i}(\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}})'$ , are used in multiple estimators. We provide details about the TM and the OM sample estimating functions here.

### TM estimating functions

The sample estimating functions used to estimate the parameters of the TM  $p(\mathbf{z}, t, \boldsymbol{\gamma})$  are the sample score equations from the quasimaximum likelihood (QML) estimator.

In the binary-treatment case,  $p(\mathbf{z}, t, \boldsymbol{\gamma})$  may be logit, probit, or heteroskedastic probit. In the multivalued-treatment case,  $p(\mathbf{z}, t, \boldsymbol{\gamma})$  is a multinomial logit. We now give formulas for the  $s_{\text{tm},i}(\mathbf{z}_i, \widehat{\boldsymbol{\gamma}})$  for each case.

### logit and probit

In the logit and probit cases,

$$s_{\text{tm},i}(\mathbf{z}_i, \widehat{\boldsymbol{\gamma}}) = \left[ \frac{g(\mathbf{z}_i \widehat{\boldsymbol{\gamma}}') \{t_i - G(\mathbf{z}_i \widehat{\boldsymbol{\gamma}}')\}}{G(\mathbf{z}_i \widehat{\boldsymbol{\gamma}}') \{1 - G(\mathbf{z}_i \widehat{\boldsymbol{\gamma}}')\}} \right] \mathbf{z}_i$$

where  $G(z)$  is the logistic cumulative distribution function for the logit,  $G(z)$  is the normal cumulative distribution function for the probit, and  $g(\cdot) = \{\partial G(z)\}/(\partial z)$  is the corresponding density function.

**hetprobit**

In the `hetprobit` case, there are two sets of sample score equations,  $\mathbf{s}_{\text{tm},1,i}(\mathbf{z}_i, \hat{\gamma})$  and  $\mathbf{s}_{\text{tm},2,i}(\mathbf{z}_i, \hat{\gamma})$ :

$$\mathbf{s}_{\text{tm},1,i}(\mathbf{z}_i, \hat{\gamma}) = \left( \frac{\phi \{q(\mathbf{z}_i, \hat{\gamma})\} [t_i - \Phi \{q(\mathbf{z}_i, \hat{\gamma})\}]}{\Phi \{q(\mathbf{z}_i, \hat{\gamma})\} [1 - \Phi \{q(\mathbf{z}_i, \hat{\gamma})\}] \exp(\hat{\mathbf{z}}_i \hat{\gamma}')}\right) \hat{\mathbf{z}}_i'$$

and

$$\mathbf{s}_{\text{tm},2,i}(\mathbf{z}_i, \hat{\gamma}) = \left( \frac{\phi \{q(\mathbf{z}_i, \hat{\gamma})\} \hat{\mathbf{z}}_i \hat{\gamma}' [\Phi \{q(\mathbf{z}_i, \hat{\gamma})\} - t_i]}{\Phi \{q(\mathbf{z}_i, \hat{\gamma})\} [1 - \Phi \{q(\mathbf{z}_i, \hat{\gamma})\}] \exp(\hat{\mathbf{z}}_i \hat{\gamma}')}\right) \hat{\mathbf{z}}_i'$$

where  $\phi(\cdot)$  is the standard normal density function, and  $q(\mathbf{z}_i, \hat{\gamma}) = (\hat{\mathbf{z}}_i \hat{\gamma}' / \exp(\hat{\mathbf{z}}_i \hat{\gamma}'))$ .

**mlogit**

In the `mlogit` case,  $p(\mathbf{z}, t, \gamma) = \exp(\mathbf{z}\gamma_t) / \{1 + \sum_{k=1}^q \exp(\mathbf{z}\gamma_k)\}$ . We present formulas for the case with treatment level 0 as the base with  $\gamma_0 = \mathbf{0}'$ ; see [R] `mlogit` for background.

There are  $q$  vectors of sample estimating functions for the `mlogit` case,  $\mathbf{s}_{\text{tm},k,i}(\mathbf{z}_i, \hat{\gamma})$  for each  $k \in \{1, \dots, q\}$ , 1 for each vector  $\hat{\gamma}_k$ ,  $k \in \{1, \dots, q\}$ . These sample estimating functions are

$$\mathbf{s}_{\text{tm},k,i}(\mathbf{z}_i, \hat{\gamma}) = \begin{cases} \{1 - p(\mathbf{z}_i, k, \hat{\gamma})\} \mathbf{z}_i' & t_i = k \\ -p(\mathbf{z}_i, k, \hat{\gamma}) \mathbf{z}_i' & \text{otherwise} \end{cases}$$

**OM estimating functions**

The parameters of the OM  $\mu(\mathbf{x}, t, \beta_t)$  are estimated either by weighted QML or by weighted nonlinear least squares. The estimating functions used to estimate the parameters of the OM are either the score equations from the weighted QML estimator or the moment conditions for the weighted nonlinear least-squares estimator.

The estimating functions for the OM parameters in  $\mu(\mathbf{x}, t, \beta_t)$  vary over the models for the conditional mean because  $\mu(\mathbf{x}, t, \beta_t)$  may be linear, logit, probit, heteroskedastic probit, or poisson.

Let  $N_t$  be the number of observations in treatment level  $t$ , and let  $t_i(t) = 1$  if  $t_i = t$ , with  $t_i(t) = 0$  if  $t_i \neq t$ .

There are two sets of sample estimating functions for the OM parameters with weights  $w_i(t)$ :

1.  $\mathbf{s}_{\text{ml,om},i}\{\mathbf{x}_i, w_i(t), \hat{\beta}_t\}$  are the sample estimating functions for the weighted QML estimator.
2.  $\mathbf{s}_{\text{nls,om},i}\{\mathbf{x}_i, w_i(t), \hat{\beta}_t\}$  are the sample estimating functions for the weighted nonlinear least-squares estimator.

**OM QML**

Here are the formulas for the  $\mathbf{s}_{\text{ml,om},i}\{\mathbf{x}_i, w_i(t), \hat{\beta}_t\}$  for each functional form choice.

**linear**

In the linear case,

$$s_{\text{ml,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t)(y_i - \mathbf{x}_i\widehat{\boldsymbol{\beta}}_t)\mathbf{x}'_i$$

**logit, flogit, probit, and fprobit**

In the logit, flogit, probit, and fprobit cases,

$$s_{\text{ml,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t) \left[ \frac{g(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t) \{y_i - G(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t)\}}{G(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t) \{1 - G(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t)\}} \right] \mathbf{x}_i$$

where  $G(z)$  is the logistic cumulative distribution function for the logit and flogit,  $G(z)$  is the normal cumulative distribution function for the probit and fprobit, and  $g(\cdot) = \{\partial G(z)/(\partial z)\}$  is the corresponding density function.

**hetprobit and fheterprobit**

In the `hetprobit` and `fheterprobit` cases, there are two sets of sample score equations,  $s_{\text{ml,om},1,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}$  and  $s_{\text{ml,om},2,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}$ :

$$s_{\text{ml,om},1,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t) \left( \frac{\phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\} [y_i - \Phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\}]}{\Phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\} [1 - \Phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\}]} \exp(\dot{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}_t') \right) \dot{\mathbf{x}}_i'$$

and

$$s_{\text{ml,om},2,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t) \left( \frac{\phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\} \dot{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}_t' [\Phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\} - y_i]}{\Phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\} [1 - \Phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\}]} \exp(\dot{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}_t') \right) \dot{\mathbf{x}}_i'$$

where  $\phi(\cdot)$  is the standard normal density function,  $s_{\text{ml,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}' = [s_{\text{ml,om},1,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}', s_{\text{ml,om},2,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}']$ , and  $q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t) = \left( \dot{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}_t' / \exp(\dot{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}_t') \right)$ .

**poisson**

In the poisson case,

$$s_{\text{ml,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t)\{y_i - \exp(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t')\}\mathbf{x}'_i$$

**OM WNL**

Here are the formulas for the  $s_{\text{nls,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}$  for each functional form choice.

**linear**

In the linear case,

$$\mathbf{s}_{\text{nls,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t)(y_i - \mathbf{x}_i\widehat{\boldsymbol{\beta}}_t)\mathbf{x}_i'$$

**logit, flogit, probit, and fprobit**

In the logit, flogit, probit, and fprobit cases,

$$\mathbf{s}_{\text{nls,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t) \left[ g(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t) \left\{ y_i - G(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t) \right\} \right] \mathbf{x}_i$$

where  $G(z)$  is the logistic cumulative distribution function for the logit and flogit,  $G(z)$  is the normal cumulative distribution function for the probit and fprobit, and  $g(\cdot) = \{\partial G(z)\}/(\partial z)$  is the corresponding density function.

**hetprobit and fheterprobit**

In the `hetprobit` and `fheterprobit` cases, there are two sets of sample score equations,  $\mathbf{s}_{\text{nls,om},1,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}$  and  $\mathbf{s}_{\text{nls,om},2,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}$ :

$$\mathbf{s}_{\text{nls,om},1,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t) \left( \frac{\phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\}}{\exp(\ddot{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}_t)} \left[ y_i - \Phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\} \right] \right) \dot{\mathbf{x}}_i'$$

and

$$\mathbf{s}_{\text{nls,om},2,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t) \left( \frac{\phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\}}{\exp(\ddot{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}_t)} \dot{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}_t' \left[ \Phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\} - y_i \right] \right) \ddot{\mathbf{x}}_i'$$

where  $\phi(\cdot)$  is the standard normal density function,  $\mathbf{s}_{\text{nls,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}' = [\mathbf{s}_{\text{nls,om},1,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}', \mathbf{s}_{\text{nls,om},2,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}']$ , and  $q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t) = \left( \dot{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}_t' / \exp(\ddot{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}_t) \right)$ .

**poisson**

In the poisson case,

$$\mathbf{s}_{\text{nls,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t)\{y_i - \exp(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t)\} \exp(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t)\mathbf{x}_i'$$

**Effect estimating functions**

We now describe the sample estimating functions for the effect parameters, which vary over estimator and effect parameter.

**RA estimators**

RA estimators estimate the effect parameters using means of the observation-level predictions of the conditional means of the outcomes. There is no model for the conditional probability of treatment.

The RA estimators use unweighted QML estimators to estimate the parameters of the conditional mean model. In other words, the RA estimators use the sample estimating functions  $s_{\text{ml,om},i}(\mathbf{x}_i, 1, \widehat{\beta})$ , given above.

For the RA estimators, the vector of sample estimating functions is the concatenation of the sample estimating functions for the effect parameters with the sample estimating functions for the OM parameters. Algebraically,

$$s_{\text{ra},i}(\mathbf{x}_i, \widehat{\theta})' = s_{\text{ra},e,i}(\mathbf{x}_i, \widehat{\theta}, \widehat{\beta})', s_{\text{ml,om},i}(\mathbf{x}_i, 1, \widehat{\beta})'$$

The estimating functions  $s_{\text{ra},e,i}(\mathbf{x}_i, \widehat{\theta}, \widehat{\beta})'$  vary over the effect parameter.

**RA for POM**

The RA estimators for the POM parameters estimate  $\theta' = (\alpha', \beta')$  using two types of estimating equations: 1) those for the POM parameters  $\alpha$ , and 2) those for the conditional-mean model parameters  $\beta_t$  in  $\mu(\mathbf{x}, t, \beta_t)$ .

The sample estimating functions for the  $\widehat{\beta}_t$  are given in *OM estimating functions* above.

The elements of  $s_{\text{ra},e,i}(\mathbf{x}_i, \widehat{\alpha}, \widehat{\beta})$  for the POM parameters  $\alpha$  are given by

$$\mu(\mathbf{x}_i, t, \widehat{\beta}_t) - \widehat{\alpha}_t \tag{RAPOM}$$

**RA for ATE**

The RA estimators for the ATE parameters estimate  $\theta' = (\tau', \beta')$  using two types of estimating equations: 1) those for the ATE parameters  $\tau$ , and 2) those for the OM parameters  $\beta_t$  in  $\mu(\mathbf{x}, t, \beta_t)$ .

The sample estimating functions that determine the  $\widehat{\beta}_t$  are given in *OM estimating functions* with  $w_i(t) = 1$ .

The elements of  $s_{\text{ra},e,i}(\mathbf{x}_i, \widehat{\tau}, \widehat{\beta})$  for the ATE parameters  $\tau$  are given by

$$\mu(\mathbf{x}_i, t, \widehat{\beta}_t) - \mu(\mathbf{x}_i, 0, \widehat{\beta}_t) - \widehat{\tau}_t \tag{RAATE}$$

**RA for ATET**

The RA estimators for the ATET parameters estimate  $\theta' = (\delta', \beta')$  using two types of estimating equations: 1) those for the ATET parameters  $\delta$ , and 2) those for the OM parameters  $\beta_t$  in  $\mu(\mathbf{x}, t, \beta_t)$ .

The sample estimating functions that determine the  $\widehat{\beta}_t$  are given in *OM estimating functions* above with  $w_i(t) = 1$ .

The elements of  $s_{\text{ra},e,i}(\mathbf{x}_i, \widehat{\delta}, \widehat{\beta})$  for the ATET parameters  $\delta$  are given by

$$Nt_i(\tilde{t})/N_{\tilde{t}} \left\{ \mu(\mathbf{x}_i, t, \widehat{\beta}_t) - \mu(\mathbf{x}_i, 0, \widehat{\beta}_t) - \widehat{\delta}_t \right\} \tag{RAATET}$$

## IPW estimators

IPW estimators estimate the effect parameters using means of the observed outcomes weighted by the inverse probability of treatment. There is no outcome model. The IPW estimators use QML estimators to estimate the parameters of the conditional probability model.

The vector of estimating functions is the concatenation of the estimating functions for the effect parameters with the estimating functions for the conditional-probability parameters. The sample estimating functions used by the IPW estimators are

$$s_{\text{ipw},i}(\mathbf{x}_i, \hat{\boldsymbol{\theta}})' = s_{\text{ipw},e,i}(\mathbf{x}_i, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\gamma}})', s_{\text{tm},i}(\mathbf{z}_i, 1, \hat{\boldsymbol{\gamma}})'$$

The estimating functions  $s_{\text{ipw},e,i}(\mathbf{z}_i, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\gamma}})'$  vary over the effect parameter.

All the IPW estimators use normalized inverse-probability weights. These weights are not related to the weights  $w_i(t)$  that were used in the OM equations. The functional form for the normalized inverse-probability weights varies over the effect parameters POM, ATE, and ATET.

The POM and ATE estimators use normalized inverse-probability weights. The unnormalized weights for individual  $i$  and treatment level  $t$  are  $d_i(t) = t_i(t)/p(\mathbf{z}_i, t, \hat{\boldsymbol{\gamma}})$ , and the normalized weights are  $\bar{d}_i(t) = N_t d_i(t) / \sum_i^N d_i(t)$ .

The ATET estimator uses normalized treatment-adjusted inverse-probability weights. The treatment-adjusted inverse-probability weights adjust the inverse-probability weights for the probability of getting the conditional treatment  $\tilde{t}$ . The unnormalized weights are  $f_i = p(\mathbf{z}_i, \tilde{t}, \hat{\boldsymbol{\gamma}})/p(\mathbf{z}_i, t_i, \hat{\boldsymbol{\gamma}})$ , and the normalized weights are  $\bar{f}_i = N f_i / \sum_i^N f_i$ .

The IPW effect estimators are weighted cell averages. The sample estimating functions used in POM estimators are the sample estimating functions from weighted OLS regression on treatment-cell indicators. The POM estimators use a full set of  $q + 1$  of treatment indicator variables  $\mathbf{a}_i$ . (The  $i$ th observation on the  $k$ th variable in  $\mathbf{a}_i$  is 1 if  $i$  received treatment  $k - 1$  and 0 otherwise, for  $k \in \{1, 2, \dots, q + 1\}$ .)

The sample estimating functions used in the ATE and the ATET estimators are the sample estimating functions from weighted OLS regression on treatment-cell indicators, excluding the indicator for the control level and including a constant term. The variables  $\tilde{\mathbf{a}}_i$  used in the ATE and ATET sample estimating functions include  $q$  of treatment indicator variables and a variable that is always 1. (The first  $q$  variables in  $\tilde{\mathbf{a}}_i$  are treatment indicators: the  $i$ th observation on the  $k$ th variable in  $\tilde{\mathbf{a}}_i$  is 1 if  $i$  received treatment  $k$  and 0 otherwise, for  $k \in \{1, 2, \dots, q\}$ . The  $(q + 1)$ th variable is always 1.) This definition of  $\tilde{\mathbf{a}}_i$  sets the last treatment level to be the control; renaming the treatments handles the more general case allowed for by `teffects`.

Having defined  $\mathbf{a}_i$  and  $\tilde{\mathbf{a}}_i$ , we can give the sample estimating functions that the IPW estimators use for the effects parameters.

## IPW for POM

We begin with the IPW estimators for the potential-outcome means. In this case,  $\boldsymbol{\theta}' = (\boldsymbol{\alpha}', \boldsymbol{\gamma}')$ .

The sample estimating functions for the  $\hat{\boldsymbol{\gamma}}$  are given in [TM estimating functions](#) above.

The sample estimating functions for  $\hat{\boldsymbol{\alpha}}$  are given by

$$s_{\text{ipw},e,i,t}(\mathbf{z}_i, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\gamma}})' = \bar{d}_i(t)(y_i - \mathbf{a}_i \hat{\boldsymbol{\alpha}}) \mathbf{a}_i' \quad (\text{IPWPOM})$$

**IPW for ATE**

The full parameter vector for the IPW estimators for the ATE is  $\theta' = (\tau', \gamma')$ .

The sample estimating functions for the  $\hat{\gamma}$  are given in *TM estimating functions* above.

The sample estimating functions for  $\hat{\tau}$  are given by

$$s_{\text{ipw},e,i,t}(\mathbf{z}_i, \hat{\tau}, \hat{\gamma})' = \bar{d}_i(t)(y_i - \tilde{\mathbf{a}}_i \hat{\tau}) \tilde{\mathbf{a}}_i' \quad (\text{IPWATE})$$

**IPW for ATET**

The full parameter vector for the IPW estimators for the ATET is  $\theta' = (\delta', \gamma')$ .

The sample estimating functions for the  $\hat{\gamma}$  are given in *TM estimating functions* above.

The sample estimating functions for  $\hat{\delta}$  are given by

$$s_{\text{ipw},e,i,t}(\mathbf{z}_i, \hat{\delta}, \hat{\gamma})' = \bar{f}_i(t)(y_i - \tilde{\mathbf{a}}_i \hat{\delta}) \tilde{\mathbf{a}}_i' \quad (\text{IPWATET})$$

**AIPW estimators**

This section documents the sample estimating functions used by the augmented inverse-probability-weighted (AIPW) estimators implemented in **teffects**. These AIPW estimators are efficient-influence-function estimators as discussed in [TE] **teffects intro** and [TE] **teffects intro advanced**. The **teffects** implementation was primarily inspired by Cattaneo, Drukker, and Holland (2013), which was based on Cattaneo (2010). Tan (2010) was influential by identifying the implemented weighted nonlinear least-squares estimator as having relatively good properties when both the conditional mean function and the conditional probability function are misspecified.

The AIPW estimating functions for the treatment parameters include terms from a conditional probability model and from a conditional mean model for the outcome.

The sample-estimation-equations vector has three parts for the AIPW estimators:

$$s_{\text{aipw},i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta})' = [s_{\text{aipw},e,i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta})', s_{\text{aipw},\text{tm},i}(\mathbf{z}_i, \hat{\gamma})', s_{\text{aipw},\text{om},i}(\mathbf{x}_i, w_i(t), \hat{\beta})']$$

The sample estimating functions for the parameters of the TM are the  $s_{\text{tm},i}(\mathbf{z}_i, \hat{\gamma})$  given in *TM estimating functions* above.

**teffects aipw** implements three AIPW estimators: the standard AIPW estimator, the NLS AIPW estimator, and the WNLS AIPW estimator.

The three AIPW estimators differ in how they estimate the parameters of the OM.

By default, **teffects aipw** uses the standard AIPW estimator that estimates the parameters of the OM by the unweighted ML estimator, whose sample estimating functions,  $s_{\text{ml},\text{om},i}(\mathbf{x}_i, 1, \hat{\beta})$ , are given in *OM estimating functions* above.

When the **nls** option is specified, **teffects aipw** uses the NLS AIPW estimator that estimates the parameters of the OM by the unweighted NLS estimator, whose sample estimating functions,  $s_{\text{nls},\text{om},i}(\mathbf{x}_i, 1, \hat{\beta})$ , are given in *OM estimating functions* above.



When the `wnls` option is specified, `teffects aipw` uses the WNLS AIPW estimator that estimates the parameters of the OM by the WNLS estimator, whose sample estimating functions,  $\mathbf{s}_{\text{wnls,om},i}\{\mathbf{x}_i, \tilde{w}_i(t), \hat{\boldsymbol{\beta}}\}$ , are given in *OM estimating functions* above. The weights for person  $i$  in treatment level  $t$  are

$$\tilde{w}_i(t) = \frac{t_i(t)}{p(\mathbf{z}_i, t, \hat{\gamma})} \left\{ \frac{t_i(t)}{p(\mathbf{z}_i, t, \hat{\gamma})} - 1 \right\} \quad (\text{WNLSW})$$

Now we discuss the sample estimating functions for the effect parameters, the  $\mathbf{s}_{e,i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\boldsymbol{\theta}})$ .

### AIPW for POM

We begin with the AIPW estimators for the potential-outcome means. In this case,  $\boldsymbol{\theta}' = (\boldsymbol{\alpha}', \boldsymbol{\gamma}', \boldsymbol{\beta}')$ , and the elements of  $\mathbf{s}_{\text{aipw},e,i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\boldsymbol{\theta}})$  are given by

$$\frac{t_i(t)}{p(\mathbf{z}_i, t, \hat{\gamma})} y_i - \mu(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_t) \left\{ \frac{t_i(t)}{p(\mathbf{z}_i, t, \hat{\gamma})} - 1 \right\} - \alpha_t$$

### AIPW for ATE

The AIPW estimators for the ATE estimate  $\boldsymbol{\theta}' = (\boldsymbol{\tau}', \boldsymbol{\gamma}', \boldsymbol{\beta}')$ , and the elements of  $\mathbf{s}_{\text{aipw},e,i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\boldsymbol{\theta}})$  are given by

$$\begin{aligned} & \frac{t_i(t)}{p(\mathbf{z}_i, t, \hat{\gamma})} y_i - \mu(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_t) \left\{ \frac{t_i(t)}{p(\mathbf{z}_i, t, \hat{\gamma})} - 1 \right\} - \tau_0 \text{ if } t = 0 \\ & \frac{t_i(t)}{p(\mathbf{z}_i, t, \hat{\gamma})} y_i - \mu(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_t) \left\{ \frac{t_i(t)}{p(\mathbf{z}_i, t, \hat{\gamma})} - 1 \right\} - \tau_t - \tau_0 \text{ if } t > 0 \end{aligned}$$

### IPWRA estimators

The IPWRA estimators combine inverse-probability weighting (IPW) with regression-adjustment (RA) methods. The sample estimating functions for IPWRA include sample estimating functions from both RA and IPW. The vector of sample estimating functions is

$$\mathbf{s}_{\text{ipwra},i}(\mathbf{x}_i, \hat{\boldsymbol{\theta}})' = \mathbf{s}_{\text{ra},e,i}(\mathbf{x}_i, \hat{\boldsymbol{\vartheta}}, \hat{\boldsymbol{\beta}})', \mathbf{s}_{\text{ml,om},i}\{\mathbf{x}_i, w_i(j), \hat{\boldsymbol{\beta}}\}', \mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \hat{\gamma})'$$

where  $\hat{\boldsymbol{\theta}}' = (\hat{\boldsymbol{\vartheta}}', \hat{\boldsymbol{\beta}}', \hat{\boldsymbol{\gamma}}')$ ,  $\hat{\boldsymbol{\vartheta}} = \hat{\boldsymbol{\alpha}}$  for POM,  $\hat{\boldsymbol{\vartheta}} = \hat{\boldsymbol{\tau}}_t$  for ATE, and  $\hat{\boldsymbol{\vartheta}} = \hat{\boldsymbol{\delta}}_t$  for ATET. The sample estimating functions,  $\mathbf{s}_{\text{ra},e,i}(\mathbf{x}_i, \hat{\boldsymbol{\vartheta}}, \hat{\boldsymbol{\beta}})$ , for POM, ATE, and ATET are given in equations (RAPOM), (RAATE), and (RAATET). For the OM sample estimating functions,  $\mathbf{s}_{\text{ml,om},i}(\cdot)$ , we replace the RA unity weights,  $w_i(t) = 1$ , with  $d_i(j)$  for POM or ATE and  $\bar{f}_i$  for ATET, the normalized inverse-probability weights described in *IPW estimators* above. The specific form of the OM sample estimating functions are given in *OM estimating functions* above. The TM sample estimating functions are given in *TM estimating functions* above.

## References

- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.
- Huber, C. 2015. Introduction to treatment effects in Stata: Part 1. *The Stata Blog: Not Elsewhere Classified*. <http://blog.stata.com/2015/07/07/introduction-to-treatment-effects-in-stata-part-1/>.
- Newey, W. K., and D. L. McFadden. 1994. Large sample estimation and hypothesis testing. In Vol. 4 of *Handbook of Econometrics*, ed. R. F. Engle and D. L. McFadden, 2111–2245. Amsterdam: Elsevier. [https://doi.org/10.1016/S1573-4412\(05\)80005-4](https://doi.org/10.1016/S1573-4412(05)80005-4).
- Stefanski, L. A., and D. D. Boos. 2002. The calculus of M-estimation. *American Statistician* 56: 29–38. <https://doi.org/10.1198/000313002753631330>.
- Tan, Z. 2010. Bounded, efficient and doubly robust estimation with inverse weighting. *Biometrika* 97: 661–682. <https://doi.org/10.1093/biomet/asq035>.
- Tsiatis, A. A. 2006. *Semiparametric Theory and Missing Data*. New York: Springer.
- van der Vaart, A. W. 1998. *Asymptotic Statistics*. Cambridge: Cambridge University Press.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

## Also see

- [TE] **teffects postestimation** — Postestimation tools for teffects
- [TE] **teffects** — Treatment-effects estimation for observational data
- [TE] **teffects ipwra** — Inverse-probability-weighted regression adjustment
- [U] **20 Estimation and postestimation commands**

# Title

**teffects ipw** — Inverse-probability weighting

<a href="#">Description</a>	<a href="#">Quick start</a>	<a href="#">Menu</a>	<a href="#">Syntax</a>
<a href="#">Options</a>	<a href="#">Remarks and examples</a>	<a href="#">Stored results</a>	<a href="#">Methods and formulas</a>
<a href="#">References</a>	<a href="#">Also see</a>		

## Description

`teffects ipw` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational data by inverse-probability weighting (IPW). IPW estimators use estimated probability weights to correct for missing data on the potential outcomes. `teffects ipw` accepts a continuous, binary, count, fractional, or nonnegative outcome and allows a multivalued treatment.

See [\[TE\] teffects intro](#) or [\[TE\] teffects intro advanced](#) for more information about estimating treatment effects from observational data.

## Quick start

ATE of binary `treat2` on `y` by IPW using a logistic model of `treat2` on `x` and `w`

```
teffects ipw (y) (treat2 x w)
```

As above, but estimate ATET

```
teffects ipw (y) (treat2 x w), atet
```

As above, but estimate potential-outcome means

```
teffects ipw (y) (treat2 x w), pomeans
```

ATE of `treat2` on `y` using heteroskedastic probit for `treat2` as a function of `x` and `w`

```
teffects ipw (y) (treat2 x w, hetprobit(x w))
```

ATE for treatment levels 2 and 3 of three-valued treatment `treat3`

```
teffects ipw (y) (treat3 x w)
```

As above, and specify that `treat3 = 3` is the control level

```
teffects ipw (y) (treat3 x w), control(3)
```

Same as above, specified using the label “MyControl” corresponding to `treat3 = 3`

```
teffects ipw (y) (treat3 x w), control(MyControl)
```

## Menu

Statistics > Treatment effects > Continuous outcomes > Inverse-probability weighting (IPW)  
Statistics > Treatment effects > Binary outcomes > Inverse-probability weighting (IPW)  
Statistics > Treatment effects > Count outcomes > Inverse-probability weighting (IPW)  
Statistics > Treatment effects > Fractional outcomes > Inverse-probability weighting (IPW)  
Statistics > Treatment effects > Nonnegative outcomes > Inverse-probability weighting (IPW)

## Syntax

```
teffects ipw (ovar) (tvar tmvarlist [, tmodel noconstant]) [if] [in] [weight]
[, stat options]
```

*ovar* is a binary, count, continuous, fractional, or nonnegative outcome of interest.

*tvar* must contain integer values representing the treatment levels.

*tmvarlist* specifies the variables that predict treatment assignment in the treatment model.

<i>tmodel</i>	Description
---------------	-------------

### Model

<code>logit</code>	logistic treatment model; the default
<code>probit</code>	probit treatment model
<code>hetprobit(<i>varlist</i>)</code>	heteroskedastic probit treatment model

*tmodel* specifies the model for the treatment variable.

For multivalued treatments, only `logit` is available and multinomial logit is used.

<i>stat</i>	Description
-------------	-------------

### Stat

<code>ate</code>	estimate average treatment effect in population; the default
<code>atet</code>	estimate average treatment effect on the treated
<code>pomeans</code>	estimate potential-outcome means

<i>options</i>	Description
----------------	-------------

### SE/Robust

<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>robust</code> , <code>cluster <i>clustvar</i></code> , <code>bootstrap</code> , or <code>jackknife</code>
----------------------------------	---

### Reporting

<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>aequations</code>	display auxiliary-equation results
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

### Maximization

<code>maximize_options</code>	control the maximization process; seldom used
-------------------------------	---

### Advanced

<code>pstolerance(#)</code>	set tolerance for overlap assumption
<code>osample(<i>newvar</i>)</code>	<i>newvar</i> identifies observations that violate the overlap assumption
<code>control(#   <i>label</i>)</code>	specify the level of <i>tvar</i> that is the control
<code>tlevel(#   <i>label</i>)</code>	specify the level of <i>tvar</i> that is the treatment
<code>coeflegend</code>	display legend instead of statistics

*tmvarlist* may contain factor variables; see [U] 11.4.3 **Factor variables**.

*bootstrap*, *by*, *collect*, *jackknife*, and *statsby* are allowed; see [U] 11.1.10 **Prefix commands**.

Weights are not allowed with the *bootstrap* prefix; see [R] **bootstrap**.

*fweights*, *iwweights*, and *pweights* are allowed; see [U] 11.1.6 **weight**.

*coeflegend* does not appear in the dialog box.

See [U] 20 **Estimation and postestimation commands** for more capabilities of estimation commands.

## Options

### Model

*noconstant*; see [R] **Estimation options**.

### Stat

*stat* is one of three statistics: *ate*, *atet*, or *pomeans*. *ate* is the default.

*ate* specifies that the average treatment effect be estimated.

*atet* specifies that the average treatment effect on the treated be estimated.

*pomeans* specifies that the potential-outcome means for each treatment level be estimated.

### SE/Robust

*vce(vctype)* specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (*robust*), that allow for intragroup correlation (*cluster clustvar*), and that use bootstrap or jackknife methods (*bootstrap*, *jackknife*); see [R] **vce\_option**.

### Reporting

*level(#)*; see [R] **Estimation options**.

*aequations* specifies that the results for the outcome-model or the treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

*display\_options*: *nocl*, *nopvalues*, *noomitted*, *vsquish*, *noemptycells*, *baselevels*, *allbaselevels*, *nofvlabel*, *fvwrap(#)*, *fvwrapon(style)*, *cformat(%fmt)*, *pformat(%fmt)*, *sformat(%fmt)*, and *nolstretch*; see [R] **Estimation options**.

### Maximization

*maximize\_options*: *iterate(#)*, *[no]log*, and *from(init\_specs)*; see [R] **Maximize**. These options are seldom used.

*init\_specs* is one of

*matname* [ , *skip copy* ]

*#* [ , *# ...* ] , *copy*

### Advanced

*pstolerance(#)* specifies the tolerance used to check the overlap assumption. The default value is *pstolerance(1e-5)*. *teffects* will exit with an error if an observation has an estimated propensity score smaller than that specified by *pstolerance()*.

*osample(newvar)* specifies that indicator variable *newvar* be created to identify observations that violate the overlap assumption.

`control(# | label)` specifies the level of *tvar* that is the control. The default is the first treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with statistic `pomeans`. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(# | label)` specifies the level of *tvar* that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

The following option is available with `teffects ipw` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

## Remarks and examples

Remarks are presented under the following headings:

[Overview](#)

[Video example](#)

### Overview

IPW estimators use estimated probability weights to correct for the missing-data problem arising from the fact that each subject is observed in only one of the potential outcomes. IPW estimators use a two-step approach to estimating treatment effects:

1. They estimate the parameters of the treatment model and compute the estimated inverse-probability weights.
2. They use the estimated inverse-probability weights to compute weighted averages of the outcomes for each treatment level. The contrasts of these weighted averages provide the estimates of the ATEs. Using this weighting scheme corrects for the missing potential outcomes.

These steps produce consistent estimates of the effect parameters because the treatment is assumed to be independent of the potential outcomes after conditioning on the covariates. The overlap assumption ensures that predicted inverse-probability weights do not get too large. In fact, `teffects ipw` uses an estimation technique that implements both steps at once so that we do not need to correct the standard errors in the second step to reflect the uncertainty associated with the predicted treatment probabilities.

We will illustrate the use of `teffects ipw` by using data from a study of the effect of a mother's smoking status during pregnancy (`mbsmoke`) on infant birthweight (`bweight`) as reported by [Cattaneo \(2010\)](#). This dataset also contains information about each mother's age (`mage`), education level (`medu`), marital status (`mmarried`), whether the first prenatal exam occurred in the first trimester (`prenatal1`), and whether this baby was the mother's first birth (`fbaby`).

### ► Example 1: Estimating the ATE

We begin by using `teffects ipw` to estimate the average treatment effect of smoking on birthweight. We will use a probit model to predict treatment status, using `prenatal1`, `mmarried`, `mage`, the square of `mage`, and `fbaby` as explanatory variables:

```
. use https://www.stata-press.com/data/r17/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ipw (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu, probit)
Iteration 0: EE criterion = 4.621e-21
Iteration 1: EE criterion = 7.358e-26
Treatment-effects estimation      Number of obs      =      4,642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: probit
```

bweight	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
mbsmoke (Smoker vs Nonsmoker)	-230.6886	25.81524	-8.94	0.000	-281.2856	-180.0917
POmean						
mbsmoke Nonsmoker	3403.463	9.571369	355.59	0.000	3384.703	3422.222

The average birthweight if all mothers were to smoke would be 231 grams less than the average of 3,403 grams that would occur if none of the mothers had smoked.

◀

Sometimes, we are mainly concerned about those subjects that did in fact receive treatment, and we want to know how much the outcome changes as a result of treatment for that subpopulation. The ATET provides us with the answer. Moreover, the ATET can be estimated using weaker assumptions than are required to estimate the ATE; see [TE] [teffects intro advanced](#).

## ▶ Example 2: Estimating the ATET

```
. teffects ipw (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu, probit),
> atet
Iteration 0: EE criterion = 4.636e-21
Iteration 1: EE criterion = 6.467e-27
Treatment-effects estimation      Number of obs      =      4,642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: probit
```

bweight	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATET						
mbsmoke (Smoker vs Nonsmoker)	-225.1773	23.66458	-9.52	0.000	-271.559	-178.7955
POmean						
mbsmoke Nonsmoker	3362.837	14.20149	236.79	0.000	3335.003	3390.671

The average birthweight is 225 grams less when all the mothers who smoke do so than the average of 3,363 grams that would have occurred if none of these mothers had smoked.

We often express statistics as percentages to alleviate scaling issues and aid interpretation. In the present context, we may wish to express an ATE as a percentage of the untreated POM to gain a more intuitive measure of the effect of treatment.

### ▷ Example 3: Reporting the ATE as a percentage

Here we use the same model as in example 1, but we report the ATE as a percentage of the mean birthweight that would occur if no mothers smoke. First, we use `teffects ipw` to fit the model. We use the `coeflegend` option so that `teffects ipw` reports the names of the parameters. Then we use `nlcom` to obtain the statistic we want along with its delta-method-based standard error. We type

```
. teffects ipw (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu, probit),
> coeflegend
Iteration 0:   EE criterion = 4.621e-21
Iteration 1:   EE criterion = 7.358e-26
Treatment-effects estimation      Number of obs   =      4,642
Estimator       : inverse-probability weights
Outcome model   : weighted mean
Treatment model : probit
```

bweight	Coefficient	Legend
ATE		
mbsmoke (Smoker vs Nonsmoker)	-230.6886	_b[ATE:r1vs0.mbsmoke]
POMean		
mbsmoke Nonsmoker	3403.463	_b[POMean:0.mbsmoke]

```
. nlcom _b[ATE:r1vs0.mbsmoke] / _b[POMean:0.mbsmoke]
      _n1_1:  _b[ATE:r1vs0.mbsmoke] / _b[POMean:0.mbsmoke]
```

bweight	Coefficient	Std. err.	z	P> z	[95% conf. interval]
_n1_1	-.0677806	.0075169	-9.02	0.000	-.0825133    -.0530478

The average birthweight falls by an estimated 6.8% when every mother smokes relative to the case when no mothers smoke. We also obtain a 95% confidence interval of a 5.3% to 8.3% reduction.

## Video example

[Treatment effects: Inverse-probability weighting](#)



## Stored results

`teffects ipw` stores the following in `e()`:

### Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level $j$
<code>e(N_clust)</code>	number of clusters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(converged)</code>	1 if converged, 0 otherwise

### Macros

<code>e(cmd)</code>	<code>teffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(subcmd)</code>	<code>ipw</code>
<code>e(tmodel)</code>	<code>logit</code> , <code>probit</code> , or <code>hetprobit</code>
<code>e(stat)</code>	statistic estimated, <code>ate</code> , <code>atet</code> , or <code>pomeans</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<code>vcetype</code> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

### Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

### Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

### Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, $p$ -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r`-class command is run after the estimation command.

## Methods and formulas

`teffects ipw` implements a [smooth treatment-effects estimator](#). All smooth treatment-effects estimators are documented in [Methods and formulas](#) of [TE] `teffects aipw`.

## References

- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.

- Cerulli, G. 2014. `treatrew`: A user-written command for estimating average treatment effects by reweighting on the propensity score. *Stata Journal* 14: 541–561.
- Drukker, D. M. 2014. Using `gmm` to solve two-step estimation problems. *The Stata Blog: Not Elsewhere Classified*. <http://blog.stata.com/2014/12/08/using-gmm-to-solve-two-step-estimation-problems/>.
- Huber, C. 2015. Introduction to treatment effects in Stata: Part 1. *The Stata Blog: Not Elsewhere Classified*. <http://blog.stata.com/2015/07/07/introduction-to-treatment-effects-in-stata-part-1/>.

## Also see

- [TE] **teffects postestimation** — Postestimation tools for `teffects`
- [TE] **teffects** — Treatment-effects estimation for observational data
- [U] **20 Estimation and postestimation commands**

# Title

**teffects ipwra** — Inverse-probability-weighted regression adjustment

<a href="#">Description</a>	<a href="#">Quick start</a>	<a href="#">Menu</a>	<a href="#">Syntax</a>
<a href="#">Options</a>	<a href="#">Remarks and examples</a>	<a href="#">Stored results</a>	<a href="#">Methods and formulas</a>
<a href="#">References</a>	<a href="#">Also see</a>		

## Description

`teffects ipwra` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational data by inverse-probability-weighted regression adjustment (IPWRA). IPWRA estimators use weighted regression coefficients to compute averages of treatment-level predicted outcomes, where the weights are the estimated inverse probabilities of treatment. The contrasts of these averages estimate the treatment effects. IPWRA estimators have the double-robust property. `teffects ipwra` accepts a continuous, binary, count, fractional, or nonnegative outcome and allows a multivalued treatment.

See [\[TE\] teffects intro](#) or [\[TE\] teffects intro advanced](#) for more information about estimating treatment effects from observational data.

## Quick start

ATE of binary treatment `treat2` estimated by IPWRA using a linear model for outcome `y1` on `x1` and `x2` and a logistic model for `treat2` on `x1` and `w`

```
teffects ipwra (y1 x1 x2) (treat2 x1 w)
```

As above, but estimate the ATET

```
teffects ipwra (y1 x1 x2) (treat2 x1 w), atet
```

Probit model for binary outcome `y3`

```
teffects ipwra (y3 x1 x2, probit) (treat2 x1 w)
```

As above, but use a heteroskedastic probit model for `y3` and a probit model for `treat2`

```
teffects ipwra (y3 x1 x2, hetprobit(x1 x2)) (treat2 x1 w, probit)
```

As above, but use a fractional heteroskedastic probit model for `y4` and a probit model for `treat2`

```
teffects ipwra (y4 x1 x2, fhetsprobit(x1 x2)) (treat2 x1 w, probit)
```

ATE for each level of a three-valued treatment `treat3`

```
teffects ipwra (y1 x1 x2) (treat3 x1 w)
```

As above, and specify that `treat3 = 3` is the control level using the value label “MyControl” for 3

```
teffects ipwra (y1 x1 x2) (treat3 x1 w), control(MyControl)
```

## Menu

Statistics > Treatment effects > Continuous outcomes > Regression adjustment with IPW

Statistics > Treatment effects > Binary outcomes > Regression adjustment with IPW

Statistics > Treatment effects > Count outcomes > Regression adjustment with IPW

Statistics > Treatment effects > Fractional outcomes > Regression adjustment with IPW

Statistics > Treatment effects > Nonnegative outcomes > Regression adjustment with IPW

## Syntax

```
teffects ipwra (ovar omvarlist [, omodel noconstant])
              (tvar tmvarlist [, tmodel noconstant]) [if] [in] [weight]
              [, stat options]
```

*ovar* is a binary, count, continuous, fractional, or nonnegative outcome of interest.

*omvarlist* specifies the covariates in the outcome model.

*tvar* must contain integer values representing the treatment levels.

*tmvarlist* specifies the covariates in the treatment-assignment model.

<i>omodel</i>	Description
---------------	-------------

Model

<code>linear</code>	linear outcome model; the default
<code>logit</code>	logistic outcome model
<code>probit</code>	probit outcome model
<code>hetprobit(<i>varlist</i>)</code>	heteroskedastic probit outcome model
<code>poisson</code>	exponential outcome model
<code>flogit</code>	fractional logistic outcome model
<code>fprobit</code>	fractional probit outcome model
<code>fheterprobit(<i>varlist</i>)</code>	fractional heteroskedastic probit outcome model

*omodel* specifies the model for the outcome variable.

<i>tmodel</i>	Description
---------------	-------------

Model

<code>logit</code>	logistic treatment model; the default
<code>probit</code>	probit treatment model
<code>hetprobit(<i>varlist</i>)</code>	heteroskedastic probit treatment model

*tmodel* specifies the model for the treatment variable.

For multivalued treatments, only `logit` is available and multinomial logit is used.

<i>stat</i>	Description
-------------	-------------

Stat

<code>ate</code>	estimate average treatment effect in population; the default
<code>atet</code>	estimate average treatment effect on the treated
<code><u>pomeans</u></code>	estimate potential-outcome means

<i>options</i>	Description
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>robust</code> , <code>cluster <i>clustvar</i></code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>aequations</code>	display auxiliary-equation results
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
Advanced	
<code>pstolerance(#)</code>	set tolerance for overlap assumption
<code>osample(<i>newvar</i>)</code>	<i>newvar</i> identifies observations that violate the overlap assumption
<code>control(#   <i>label</i>)</code>	specify the level of <i>tvar</i> that is the control
<code>tlevel(#   <i>label</i>)</code>	specify the level of <i>tvar</i> that is the treatment
<code>coeflegend</code>	display legend instead of statistics

*omvarlist* and *tmvarlist* may contain factor variables; see [U] 11.4.3 Factor variables.

`bootstrap`, `by`, `collect`, `jackknife`, and `statsby` are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the `bootstrap` prefix; see [R] `bootstrap`.

`fweights`, `iweights`, and `pweights` are allowed; see [U] 11.1.6 weight.

`coeflegend` does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

## Options

### Model

`noconstant`; see [R] Estimation options.

### Stat

*stat* is one of three statistics: `ate`, `atet`, or `pomeans`. `ate` is the default.

`ate` specifies that the average treatment effect be estimated.

`atet` specifies that the average treatment effect on the treated be estimated.

`pomeans` specifies that the potential-outcome means for each treatment level be estimated.

### SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster clustvar`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] `vce_option`.

## Reporting

`level(#)`; see [R] [Estimation options](#).

`aequations` specifies that the results for the outcome-model or the treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

*display\_options*: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

## Maximization

*maximize\_options*: `iterate(#)`, `[no]log`, and `from(init_specs)`; see [R] [Maximize](#). These options are seldom used.

*init\_specs* is one of

`matname [ , skip copy ]`

`# [ , # ... ] , copy`

## Advanced

`pstolerance(#)` specifies the tolerance used to check the overlap assumption. The default value is `pstolerance(1e-5)`. `teffects` will exit with an error if an observation has an estimated propensity score smaller than that specified by `pstolerance()`.

`osample(newvar)` specifies that indicator variable *newvar* be created to identify observations that violate the overlap assumption.

`control(# | label)` specifies the level of *tvar* that is the control. The default is the first treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with statistic `pomeans`. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(# | label)` specifies the level of *tvar* that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

The following option is available with `teffects ipwra` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

## Remarks and examples

Remarks are presented under the following headings:

[Overview](#)

[Video example](#)

## Overview

IPWRA estimators use probability weights to obtain outcome-regression parameters that account for the missing-data problem arising from the fact that each subject is observed in only one of the potential outcomes. The adjusted outcome-regression parameters are used to compute averages of treatment-level predicted outcomes. The contrasts of these averages provide estimates of the treatment effects.

IPWRA estimators use a model to predict treatment status, and they use another model to predict outcomes. Because IPWRA estimators have the double-robust property, only one of the two models must be correctly specified for the IPWRA estimator to be consistent.

IPWRA estimators use a three-step approach to estimating treatment effects:

1. They estimate the parameters of the treatment model and compute inverse-probability weights.
2. Using the estimated inverse-probability weights, they fit weighted regression models of the outcome for each treatment level and obtain the treatment-specific predicted outcomes for each subject.
3. They compute the means of the treatment-specific predicted outcomes. The contrasts of these averages provide the estimates of the ATEs. By restricting the computations of the means to the subset of treated subjects, we can obtain the ATETS.

These steps produce consistent estimates of the effect parameters because the treatment is assumed to be independent of the potential outcomes after conditioning on the covariates. The overlap assumption ensures that predicted inverse-probability weights do not get too large. The standard errors reported by `teffects ipwra` correct for the three-step process. See [\[TE\] teffects intro](#) or [\[TE\] teffects intro advanced](#) for more information about this estimator.

We will illustrate the use of `teffects ipwra` by using data from a study of the effect of a mother's smoking status during pregnancy (`mbsmoke`) on infant birthweight (`bweight`) as reported by [Cattaneo \(2010\)](#). This dataset also contains information about each mother's age (`mage`), education level (`medu`), marital status (`mmarried`), whether the first prenatal exam occurred in the first trimester (`prenatal1`), and whether this baby was the mother's first birth (`fbaby`).

## ▷ Example 1: Estimating the ATE

We begin by using `teffects ipwra` to estimate the average treatment effect of smoking on birthweight. We will use a probit model to predict treatment status as a function of `mmarried`, `mage`, and `fbaby`; to maximize the predictive power of this model, we use factor-variable notation to incorporate quadratic effects of the mother's age, the only continuous covariate in our model. We will use linear regression (the default) to model birthweight, using `prenatal1`, `mmarried`, `mage`, and `fbaby` as explanatory variables. We type

```

. use https://www.stata-press.com/data/r17/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ipwra (bweight prenatal1 mmarried mage fbaby)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, probit)

Iteration 0:   EE criterion = 9.885e-21
Iteration 1:   EE criterion = 7.847e-26

Treatment-effects estimation           Number of obs   =       4,642
Estimator      : IPW regression adjustment
Outcome model  : linear
Treatment model: probit

```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
mbsmoke (Smoker vs Nonsmoker)	-229.9671	26.62668	-8.64	0.000	-282.1544	-177.7798
POMean						
mbsmoke Nonsmoker	3403.336	9.57126	355.58	0.000	3384.576	3422.095

The average birthweight if all mothers were to smoke would be 230 grams less than the average of 3,403 grams that would occur if none of the mothers had smoked.

◀

By default, `teffects ipwra` displays the ATE and untreated POM. We can specify the `pomeans` option to display both the treated and untreated POMs, and we can use the `aequations` option to display the regression model coefficients used to predict the POMs as well as the coefficients from the model used to predict treatment.



## ► Example 2: Displaying the POMs and equations

```
. use https://www.stata-press.com/data/r17/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ipwra (bweight prenatal1 mmarried mage fbaby)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, probit), pomeans aequations

Iteration 0:  EE criterion = 9.885e-21
Iteration 1:  EE criterion = 6.922e-26

Treatment-effects estimation      Number of obs   =      4,642
Estimator      : IPW regression adjustment
Outcome model  : linear
Treatment model: probit
```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
<b>POmeans</b>						
mbsmoke						
Nonsmoker	3403.336	9.57126	355.58	0.000	3384.576	3422.095
Smoker	3173.369	24.86997	127.60	0.000	3124.624	3222.113
<b>OME0</b>						
prenatal1	67.98549	28.78428	2.36	0.018	11.56933	124.4017
mmarried	155.5893	26.46903	5.88	0.000	103.711	207.4677
mage	2.893051	2.134788	1.36	0.175	-1.291056	7.077158
fbaby	-71.9215	20.39317	-3.53	0.000	-111.8914	-31.95162
_cons	3194.808	55.04911	58.04	0.000	3086.913	3302.702
<b>OME1</b>						
prenatal1	34.76923	43.18534	0.81	0.421	-49.87248	119.4109
mmarried	124.0941	40.29775	3.08	0.002	45.11193	203.0762
mage	-5.068833	5.954425	-0.85	0.395	-16.73929	6.601626
fbaby	39.89692	56.82072	0.70	0.483	-71.46966	151.2635
_cons	3175.551	153.8312	20.64	0.000	2874.047	3477.054
<b>TME1</b>						
mmarried	-.6484821	.0554173	-11.70	0.000	-.757098	-.5398663
mage	.1744327	.0363718	4.80	0.000	.1031452	.2457202
c.mage#						
c.mage	-.0032559	.0006678	-4.88	0.000	-.0045647	-.0019471
fbaby	-.2175962	.0495604	-4.39	0.000	-.3147328	-.1204595
medu	-.0863631	.0100148	-8.62	0.000	-.1059917	-.0667345
_cons	-1.558255	.4639691	-3.36	0.001	-2.467618	-.6488926

◀

As is well known, the standard probit model assumes that the error terms in the latent-utility framework are homoskedastic; the model is not robust to departures from this assumption. An alternative is to use the heteroskedastic probit model, which explicitly models the error variance as a function of a set of variables.

## ► Example 3: Heteroskedastic probit treatment model

Here we use the variables as before, but we use a heteroskedastic probit model to predict treatment status, modeling the heteroskedasticity as a quadratic function of the mother's age:

```
. teffects ipwra (bweight prenatal1 mmarried fbaby c.mage)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, hetprobit(c.mage##c.mage)),
> aequations

Iteration 0:  EE criterion = 3.904e-10
Iteration 1:  EE criterion = 4.647e-11

Treatment-effects estimation          Number of obs    =    4,642
Estimator      : IPW regression adjustment
Outcome model  : linear
Treatment model: heteroskedastic probit
```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
<b>ATE</b>						
mbsmoke (Smoker vs Nonsmoker)	-229.6322	26.33451	-8.72	0.000	-281.2468	-178.0175
<b>POMean</b>						
mbsmoke Nonsmoker	3403.74	9.545798	356.57	0.000	3385.03	3422.449
<b>OME0</b>						
prenatal1	64.95123	28.6216	2.27	0.023	8.853921	121.0485
mmarried	154.2297	26.45867	5.83	0.000	102.3717	206.0878
fbaby	-71.61131	20.33774	-3.52	0.000	-111.4725	-31.75007
mage	3.010149	2.133812	1.41	0.158	-1.172046	7.192344
_cons	3195.355	55.05451	58.04	0.000	3087.45	3303.26
<b>OME1</b>						
prenatal1	38.55275	43.57023	0.88	0.376	-46.84334	123.9488
mmarried	126.3378	40.7398	3.10	0.002	46.48921	206.1863
fbaby	45.43551	56.44827	0.80	0.421	-65.20106	156.0721
mage	-6.069917	5.952507	-1.02	0.308	-17.73662	5.596783
_cons	3195.795	152.3978	20.97	0.000	2897.101	3494.489
<b>TME1</b>						
mmarried	-.0295517	.0238767	-1.24	0.216	-.0763492	.0172459
mage	.015789	.0105411	1.50	0.134	-.0048711	.0364492
c.mage#						
c.mage	-.0002837	.0001899	-1.49	0.135	-.000656	.0000886
fbaby	-.0093304	.0079946	-1.17	0.243	-.0249996	.0063387
medu	-.0036772	.0030294	-1.21	0.225	-.0096147	.0022602
_cons	-.1822172	.117975	-1.54	0.122	-.413444	.0490095
<b>TME1_lnsigma</b>						
mage	-.2211507	.0631072	-3.50	0.000	-.3448385	-.097463
c.mage#						
c.mage	.0037613	.001243	3.03	0.002	.0013251	.0061976

The estimated ATE and base-level POM are essentially the same as those produced by the model that used a homoskedastic probit.

## Video example

Treatment effects: Inverse-probability-weighted regression adjustment

## Stored results

teffects ipwra stores the following in `e()`:

### Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level $j$
<code>e(N_clust)</code>	number of clusters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(converged)</code>	1 if converged, 0 otherwise

### Macros

<code>e(cmd)</code>	teffects
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(subcmd)</code>	ipwra
<code>e(tmodel)</code>	logit, probit, or hetprobit
<code>e(omodel)</code>	linear, logit, probit, hetprobit, poisson, flogit, fprobit, or fhetsprobit
<code>e(stat)</code>	statistic estimated, ate, atet, or pomeans
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	b V
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

### Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

### Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

### Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, $p$ -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r`-class command is run after the estimation command.

## Methods and formulas

`teffects ipwra` implements a [smooth treatment-effects estimator](#). All smooth treatment-effects estimators are documented in [Methods and formulas](#) of [TE] **teffects aipw**.

## References

- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Huber, C. 2015. Introduction to treatment effects in Stata: Part 1. *The Stata Blog: Not Elsewhere Classified*. <http://blog.stata.com/2015/07/07/introduction-to-treatment-effects-in-stata-part-1/>.

## Also see

- [TE] **teffects postestimation** — Postestimation tools for teffects
- [TE] **teffects** — Treatment-effects estimation for observational data
- [TE] **teffects aipw** — Augmented inverse-probability weighting
- [U] **20 Estimation and postestimation commands**

## Description

This entry discusses the use of `teffects` when the treatment is multivalued. This entry presumes you are already familiar with the potential-outcome framework and the use of the `teffects` commands with binary treatments. See [\[TE\] teffects intro](#) or [\[TE\] teffects intro advanced](#) for more information.

## Remarks and examples

Remarks are presented under the following headings:

*[Introduction](#)*

*[Parameters and notation](#)*

*[Illustrating multivalued treatments](#)*

*[Examples](#)*

## Introduction

When the treatment is binary, each subject could either receive the treatment or not receive the treatment. In contrast, multivalued treatments refer to cases in which each subject could receive one of several different treatments or else not receive treatment at all. For example, in testing the efficacy of a drug, a patient could receive a 10 milligram (mg) dose, a 20 mg dose, a 30 mg dose, or no dose at all. We first want to be able to compare a patient receiving the 10 mg dose with a patient receiving no dose, a patient receiving the 20 mg dose with a patient receiving no dose, and a patient receiving the 30 mg dose with a patient receiving no dose. Once we can make those comparisons, we can then, for example, compare the efficacy of a 30 mg dose with that of a 20 mg dose or a 10 mg dose.

To highlight an example in economics, we consider an unemployed person who could participate in a comprehensive skills training program, attend a one-day workshop that helps job seekers write their resumés, or choose not to participate in either. We want to know how effective each of those programs is relative to not participating; once we know that, we can then compare the effectiveness of the comprehensive program with that of the one-day program.

Multivalued treatments increase the number of parameters that must be estimated and complicate the notation. Fortunately, however, using the `teffects` commands is not much more difficult with multivalued treatments than with binary treatments.

You can use `teffects ra`, `teffects ipw`, `teffects ipwra`, and `teffects aipw` to estimate multivalued treatment effects. However, the theory developed in [Abadie and Imbens \(2006, 2012\)](#) has not been extended to handle multivalued treatments, so you cannot use `teffects nmatch` or `teffects psmatch` in these cases.

[Cattaneo \(2010\)](#), [Imbens \(2000\)](#), and [Wooldridge \(2010, sec. 21.6.3\)](#) discuss aspects of treatment-effect estimation with multivalued treatments.

## Parameters and notation

We denote the potential outcome that subject  $i$  would obtain if given treatment-level  $t$  as  $y_{ti}$ , where  $y_{ti}$  is the realization of the random variable  $y_t$ . Throughout this entry,  $i$  subscripts denote realizations of the corresponding unsubscripted random variables. We again let  $y_0$  denote the potential outcome of a subject who did not receive any treatment. To handle the case of multivalued treatments, we extend the definition of the unobservable, individual-level treatment effects to be  $y_t - y_0$  for  $t \in \{1, \dots, q\}$ .

As in the binary-valued case, we again focus on three parameters of interest: the average treatment effect (ATE), the potential-outcome mean (POM), and the average treatment effect on the treated (ATET).

**ATE** The ATE is the average effect of giving each individual treatment  $t$  instead of treatment 0:

$$\text{ATE}_t = E(y_t - y_0)$$

**POM** The POM for each treatment level is an average of each potential outcome:

$$\text{POM}_t = E(y_t)$$

**ATET** The ATET is the average effect among those subjects that receive treatment level  $\tilde{t}$  of giving each subject treatment  $\tilde{t}$  instead of treatment 0:

$$\text{ATET}_{\tilde{t}, \tilde{t}} = E\{(y_{\tilde{t}} - y_0) | t = \tilde{t}\}$$

The extra notation required to define the ATET in this case indicates the difficulties surrounding this parameter.

Defining the ATET in the multivalued treatment case requires three different treatment levels:  $\tilde{t}$  defines the treatment level of the treated potential outcome; 0 is the treatment level of the control potential outcome; and  $t = \tilde{t}$  restricts the expectation to include only those individuals who actually receive treatment level  $\tilde{t}$ .

## Illustrating multivalued treatments

To illustrate the concept of a potential outcome and the parameters we would like to estimate, we consider the following table:

$y$	$t$	$y_0$	$y_1$	$y_2$
-0.50	0	-0.50	1.06	1.93
2.42	1	2.13	2.42	2.43
3.15	2	1.26	2.57	3.15
-0.39	0	-0.39	-0.18	0.52
2.22	2	-0.24	-0.01	2.22

We observe the outcome  $y$  as well as the treatment indicator  $t$ . There are three levels of treatment: 0, 1, or 2. Ideally, we would observe  $y_0$ ,  $y_1$ , and  $y_2$ , but in fact all we have is  $y$ . In the first row, the subject received treatment level 0, so  $y = y_0$  for that subject. In the last row, the subject received treatment 2, so  $y = y_2$ . We reiterate that we do not actually observe  $y_0$ ,  $y_1$ , or  $y_2$ .

If we did have data on  $y_0$ ,  $y_1$ , and  $y_2$ , then we could define subject-level treatment variables  $te_1 = y_1 - y_0$  and  $te_2 = y_2 - y_0$ . Here we would be following the convention of taking treatment level 0 to be the control level. The following table adds these two variables:

$y$	$t$	$y_0$	$y_1$	$y_2$	$te_1$	$te_2$
-0.50	0	-0.50	1.06	1.93	1.56	2.43
2.42	1	2.13	2.42	2.43	0.29	0.30
3.15	2	1.26	2.57	3.15	1.31	1.89
-0.39	0	-0.39	-0.18	0.52	0.21	0.91
2.22	2	-0.24	-0.01	2.22	0.23	2.46

Once we have  $te_1$  and  $te_2$ , obtaining the ATEs is straightforward. The ATE of going from treatment 0 to treatment 1 is simply the mean of the five entries in the column labeled  $te_1$ , which here works out to 0.72. Going from treatment level 0 to treatment level 1 causes the outcome to increase an average of 0.72. Similarly, the ATE of going from treatment 0 to treatment 2 is the mean of the entries in the column labeled  $te_2$ , which is 1.60. Exposing all subjects to treatment level 2 would cause the outcome to rise by an average of 1.60 relative to the outcome obtained by exposing them to treatment level 0.

The ATET is the average difference in the potential outcomes among those that get a particular treatment level. To compute this, we must specify two treatment levels: the actual treatment level the subjects we are interested in received as well as the treatment level we want to compare them with. For example, suppose we are interested in the ATET of going from treatment 0 to treatment 1 for those who received treatment 0. This ATET is the average of  $te_1$  for those subjects for which  $t = 0$ . Here that ATET is just  $(1.56 + 0.21)/2 \approx 0.89$ . If we exposed the subjects who received treatment 0 to treatment 1 instead, the outcome would increase an average of 0.89.

The ATET of going from treatment 0 to treatment 2 for those subjects who received treatment 2 is the mean of  $te_2$  for those subjects for which  $t = 2$ , which is  $(1.89 + 2.46)/2 \approx 2.18$ . Receiving treatment 2 increased the outcome of those who received treatment 2 by an average of 2.18 relative to receiving the control.

## Examples

In the remainder of this entry, we provide several examples demonstrating how to estimate multivalued treatments using `teffects`.

### ► Example 1: Potential outcomes with four treatment levels

`bdsianesi5.dta` contains an extract of data from [Blundell, Dearden, and Sianesi \(2005\)](#). In this dataset on individuals in the United Kingdom, `wages` records hourly wages in pounds; `ed` records the highest educational degree obtained; `paed` records the highest educational level obtained by each individual's father; `math7` records a score obtained on a standardized math test when the individual was seven; `read7` records a score obtained on a standardized reading test when the individual was seven; and `london` and `eastern` are indicators for whether an individual lives in the expensive area of London or the east. We want to know how the level of education obtained affects a person's wage.

We begin by using `mean` to report the estimated means of wages over the four education levels. The value labels on `mean` are coded as `none` for no degree, `0` for an O-level degree, `A` for an A-level degree, or `H` for a higher-education degree.

```
. use https://www.stata-press.com/data/r17/bdsianesi5
(Excerpt from Blundell, Dearden, & Sianesi (2005) JRSSA 168: 473)
. mean wage, over(ed)
Mean estimation                               Number of obs = 1,693
```

	Mean	Std. err.	[95% conf. interval]	
c.wage@ed				
none	6.057816	.154332	5.755114	6.360518
O	7.501648	.1807359	7.147158	7.856137
A	8.220637	.1540359	7.918516	8.522758
H	10.87703	.2257888	10.43417	11.31988

The output reveals that the estimated mean wage increases as the education level goes from no degree to an O-level degree, to an A-level degree, and to a higher-education degree, as we would expect. Once we control for other characteristics of each individual, do we still observe a positive effect of education on wage?

We use `teffects ra` (see [TE] [teffects ra](#)) to estimate the ATEs of the different education levels by regression adjustment (RA), controlling for each person's location, math score, and father's education level:

```
. teffects ra (wage london eastern paed math7, poisson) (ed)
Iteration 0:  EE criterion = 1.865e-18
Iteration 1:  EE criterion = 4.077e-30
Treatment-effects estimation           Number of obs   =       1,693
Estimator      : regression adjustment
Outcome model  : Poisson
Treatment model: none
```

wage	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
ed						
(O vs none)	1.181543	.3520371	3.36	0.001	.4915626	1.871523
(A vs none)	1.743079	.3282152	5.31	0.000	1.099789	2.386369
(H vs none)	3.972829	.3840024	10.35	0.000	3.220199	4.72546
POmean						
ed						
none	6.525873	.2931933	22.26	0.000	5.951224	7.100521

Because wages are necessarily positive, we used the `poisson` option inside the outcome-model specification. The estimated POM of the control level of no degree is 6.53 pounds per hour. The estimated ATE of going from no degree to an O-level degree is 1.18 pounds per hour; the estimated ATE of going from no degree to an A-level degree is 1.74 pounds per hour; and the estimated ATE of going from no degree to a higher-education degree is 3.97 pounds per hour. All of these effects are highly significant.

For comparison purposes, we also use `teffects aipw` (see [TE] [teffects aipw](#)). We use the same outcome model as before. We use a multinomial logit model to predict education level, using math and reading scores and both the father's and the mother's educational attainment levels as predictors:



```

. teffects aipw (wage london eastern paed math7, poisson)
> (ed math7 read7 maed paed)
Iteration 0:   EE criterion = 1.877e-18
Iteration 1:   EE criterion = 1.029e-30
Treatment-effects estimation          Number of obs   =       1,693
Estimator      : augmented IPW
Outcome model  : Poisson by ML
Treatment model: (multinomial) logit

```

wage	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
ed						
(0 vs none)	1.748197	.3911167	4.47	0.000	.9816221	2.514771
(A vs none)	2.363228	.3741584	6.32	0.000	1.629891	3.096565
(H vs none)	4.359777	.4133059	10.55	0.000	3.549712	5.169842
POmean						
ed						
none	5.946184	.3391531	17.53	0.000	5.281456	6.610912

The results indicate slightly higher treatment effects relative to those indicated by `teffects ra`. That is largely because the AIPW estimator predicts a lower no-higher-education POM than the RA estimator.

◀

## ► Example 2: Expressing ATEs as percentages

As in the binary-treatment case, expressing the ATEs as percentages of the POM for the control level often aids interpretation. Here we first use the replay facility of `teffects aipw` along with the `coeflegend` option to see how the parameters are named.

```

. teffects, coeflegend
Treatment-effects estimation          Number of obs   =       1,693
Estimator      : augmented IPW
Outcome model  : Poisson by ML
Treatment model: (multinomial) logit

```

wage	Coefficient	Legend
ATE		
ed		
(0 vs none)	1.748197	_b[ATE:r1vs0.ed]
(A vs none)	2.363228	_b[ATE:r2vs0.ed]
(H vs none)	4.359777	_b[ATE:r3vs0.ed]
POmean		
ed		
none	5.946184	_b[POmean:0.ed]

Now that we know the names, we can use `nlcom` to obtain the ATEs relative to the base-level POM:

```
. nlcom (_b[ATE:r1vs0.ed] / _b[P0mean:0.ed])
>      (_b[ATE:r2vs0.ed] / _b[P0mean:0.ed])
>      (_b[ATE:r3vs0.ed] / _b[P0mean:0.ed])

      _nl_1: _b[ATE:r1vs0.ed] / _b[P0mean:0.ed]
      _nl_2: _b[ATE:r2vs0.ed] / _b[P0mean:0.ed]
      _nl_3: _b[ATE:r3vs0.ed] / _b[P0mean:0.ed]
```

wage	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
_nl_1	.2940031	.0808156	3.64	0.000	.1356075	.4523988
_nl_2	.3974361	.0840545	4.73	0.000	.2326923	.5621799
_nl_3	.7332059	.1068848	6.86	0.000	.5237156	.9426962

Wages are 29% higher when everyone receives an O-level degree than if no one receives a degree.  
 Wages are 40% higher when everyone receives an A-level degree than if no one receives a degree.  
 Wages are 73% higher when everyone receives an H-level degree than if no one receives a degree.

Although impressive, these changes are not presented in the way that is most commonly discussed. (There is a large amount of literature on the treatment effect of getting a higher-education degree.) In particular, we might rather want to know the percentage changes in wages relative to a person with an A-level degree. Next we estimate the ATEs treating an A-level degree as the control level; to do that, we use the `control()` option. We also specify `coeflegend` again because we are more interested in how the parameters are named rather than in their standard errors at this point:

```
. teffects aipw (wage london eastern paed math7, poisson)
> (ed math7 read7 maed paed), control(A) coeflegend

Iteration 0:  EE criterion = 1.870e-18
Iteration 1:  EE criterion = 2.882e-30

Treatment-effects estimation          Number of obs    =    1,693
Estimator      : augmented IPW
Outcome model  : Poisson by ML
Treatment model: (multinomial) logit
```

wage	Coefficient	Legend
ATE		
ed		
(none vs A)	-2.363228	_b[ATE:r0vs2.ed]
(O vs A)	-.6150312	_b[ATE:r1vs2.ed]
(H vs A)	1.996549	_b[ATE:r3vs2.ed]
P0mean		
ed		
A	8.309412	_b[P0mean:2.ed]

Now we use `nlcom` to obtain the ATE of obtaining a higher-education degree as a percentage of the expected A-level wage:

```
. nlcom _b[ATE:r3vs2.ed] / _b[P0mean:2.ed], noheader
```

wage	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
_nl_1	.2402756	.0355404	6.76	0.000	.1706177	.3099335

The average wage increases by 24% when everyone receives an H-level degree relative to when everyone receives an A-level degree.

◀

### ▷ Example 3: Obtaining ATETs

In the previous example, we showed that on average, a higher-education degree increases a person's wage by 24% relative to someone with only an A-level degree. Sometimes, though, we would rather know how much the higher-education degree increases wages among the people who actually have a higher-education degree. To answer that question, we want to examine the ATET rather than the ATE.

Here we use the IPWRA estimator to obtain our answer. We specify the `control(A)` option so that an A-level education is treated as the basis for comparisons. We specify the `atet` option to obtain ATETs rather than ATEs, and we specify the `tlevel(H)` option to indicate that we want the ATETs to be calculated for the subset of people who actually receive higher-education degrees.

```
. teffects ipwra (wage london paed math7, poisson)
> (ed math7 read7 maed paed), atet control(A) tlevel(H)

Iteration 0:  EE criterion = 2.731e-18
Iteration 1:  EE criterion = 5.636e-31

Treatment-effects estimation          Number of obs    =      1,693
Estimator      : IPW regression adjustment
Outcome model  : Poisson
Treatment model: (multinomial) logit
```

		Robust				[95% conf. interval]	
	wage	Coefficient	std. err.	z	P> z		
ATET							
	ed						
	(none vs A)	-2.87423	.361093	-7.96	0.000	-3.58196	-2.166501
	(O vs A)	-.8246604	.3609131	-2.28	0.022	-1.532037	-.1172837
	(H vs A)	1.866757	.3277701	5.70	0.000	1.224339	2.509174
P0mean							
	ed						
	A	9.010271	.2503971	35.98	0.000	8.519501	9.50104

The point estimates are similar to the ATEs we obtained above, suggesting that the means of the covariates among those with a higher-education degree are similar to the means for the entire population.

In output not shown to save space, we replayed the previous results with the `coeflegend` option to determine how the parameters are named. Armed with that information, we call `nlcom`:

```
. nlcom _b[ATET:r3vs2.ed] / _b[P0mean:2.ed], noheader
```

	wage	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
	_nl_1	.207181	.0407528	5.08	0.000	.127307	.287055

Our estimate of the percentage increase is now noticeably smaller once we restrict ourselves to only those people who actually received a higher-education degree. However, because of the width of the confidence intervals, there is no evidence to suggest that the difference between the estimates is statistically significant.

◀

### ► Example 4: ATEs comparing adjacent treatments

In the [first example](#), we obtained the three ATEs, and they were all expressed relative to the base level of no degree. Now we show how we can express the gains to an O-level degree relative to no degree, the gains to an A-level degree relative to an O-level degree, and the gains to a higher-education degree relative to an A-level degree.

First, we use an AIPW estimator to obtain all the POMs for our example dataset:

```
. teffects aipw (wage london eastern paed math7, poisson)
> (ed math7 read7 maed paed), pom
Iteration 0:   EE criterion = 1.877e-18
Iteration 1:   EE criterion = 1.542e-30
Treatment-effects estimation           Number of obs   =       1,693
Estimator      : augmented IPW
Outcome model  : Poisson by ML
Treatment model: (multinomial) logit
```

wage	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
POMeans						
ed						
none	5.946184	.3391531	17.53	0.000	5.281456	6.610912
O	7.694381	.1915192	40.18	0.000	7.31901	8.069752
A	8.309412	.1563348	53.15	0.000	8.003001	8.615823
H	10.30596	.2285837	45.09	0.000	9.857945	10.75398

ATEs are contrasts of POMs, and here we show how to use `contrast` to obtain the estimated ATEs:

```
. contrast r.ed, nowald
warning: cannot perform check for estimable functions.
Contrasts of marginal linear predictions
Margins: asbalanced
```

	Contrast	Std. err.	[95% conf. interval]	
POMeans				
ed				
(O vs none)	1.748197	.3911167	.9816221	2.514771
(A vs none)	2.363228	.3741584	1.629891	3.096565
(H vs none)	4.359777	.4133059	3.549712	5.169842

These estimated ATEs match those we obtained in [example 2](#).

Now that we know how to use `contrast` to obtain the ATEs based on the POMs, we can take advantage of `contrast`'s ability to obtain “reverse adjacent” contrasts, which compare each level with the previous level. We use the `ar.` operator with `contrast` to accomplish this:

```
. contrast ar.ed, nowald
warning: cannot perform check for estimable functions.
Contrasts of marginal linear predictions
Margins: asbalanced
```

	Contrast	Std. err.	[95% conf. interval]	
P0means				
ed				
(0 vs none)	1.748197	.3911167	.9816221	2.514771
(A vs 0)	.6150312	.2432806	.13821	1.091852
(H vs A)	1.996549	.2730712	1.461339	2.531759

These ATEs are for incremental increases. In contrast, the ATEs considered above had a common base.

◀

## □ Technical note

The multivalued treatment AIPW estimators implemented in `teffects aipw` are EIF estimators based on the results of Cattaneo (2010). The results in Cattaneo (2010) are for semiparametric estimators, and we implement parametric versions. Of more practical importance, Cattaneo (2010) contains results for quantile treatment effects that are not implemented in `teffects` but implemented in the community-contributed `poparms` command discussed in Cattaneo, Drukker, and Holland (2013). See Emsley et al. (2008) for another implementation of the AIPW estimator, and see Frölich and Melly (2010) for other commands that estimate quantile treatment effects.

□

## References

- Abadie, A., and G. W. Imbens. 2006. Large sample properties of matching estimators for average treatment effects. *Econometrica* 74: 235–267. <https://doi.org/10.1111/j.1468-0262.2006.00655.x>.
- . 2012. Matching on the estimated propensity score. Harvard University and National Bureau of Economic Research. <http://www.nber.org/papers/w15301>.
- Blundell, R. W., L. Dearden, and B. Sianesi. 2005. Evaluating the effect of education on earnings: Models, methods and results from the National Child Development Survey. *Journal of the Royal Statistical Society, Series A* 168: 473–512. <https://doi.org/10.1111/j.1467-985X.2004.00360.x>.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.
- Emsley, R., M. Lunt, A. Pickles, and G. Dunn. 2008. Implementing double-robust estimators of causal effects. *Stata Journal* 8: 334–353.
- Frölich, M., and B. Melly. 2010. Estimation of quantile treatment effects with Stata. *Stata Journal* 10: 423–457.
- Imbens, G. W. 2000. The role of the propensity score in estimating dose–response functions. *Biometrika* 87: 706–710. <https://doi.org/10.1093/biomet/87.3.706>.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

## Also see

[TE] [teffects](#) — Treatment-effects estimation for observational data

[TE] [teffects intro](#) — Introduction to treatment effects for observational data

[TE] [teffects intro advanced](#) — Advanced introduction to treatment effects for observational data

# Title

**teffects nnmatch** — Nearest-neighbor matching

<a href="#">Description</a>	<a href="#">Quick start</a>	<a href="#">Menu</a>	<a href="#">Syntax</a>
<a href="#">Options</a>	<a href="#">Remarks and examples</a>	<a href="#">Stored results</a>	<a href="#">Methods and formulas</a>
<a href="#">References</a>	<a href="#">Also see</a>		

## Description

`teffects nnmatch` estimates the average treatment effect (ATE) and average treatment effect on the treated (ATET) from observational data by nearest-neighbor matching (NNM). NNM estimators impute the missing potential outcome for each subject by using an average of the outcomes of similar subjects that receive the other treatment level. Similarity between subjects is based on a weighted function of the covariates for each observation. The treatment effect is computed by taking the average of the difference between the observed and imputed potential outcomes for each subject. `teffects nnmatch` accepts a continuous, binary, count, fractional, or nonnegative outcome.

See [\[TE\] teffects intro](#) or [\[TE\] teffects intro advanced](#) for more information about estimating treatment effects from observational data.

## Quick start

ATE of `treat` on `y` estimated by NNM on `x1` and [indicators](#) for levels of categorical variable `a`

```
teffects nnmatch (y x1 i.a) (treat)
```

As above, but estimate the ATET

```
teffects nnmatch (y x1 i.a) (treat), atet
```

Add continuous covariate `x2` and perform bias correction

```
teffects nnmatch (y x1 x2 i.a) (treat), biasadj(x1 x2)
```

As above, and match exactly on values of `a`

```
teffects nnmatch (y x1 x2 i.a) (treat), biasadj(x1 x2) ematch(i.a)
```

With robust standard errors

```
teffects nnmatch (y x1 x2 i.a) (treat), vce(robust)
```

With four matches per observation

```
teffects nnmatch (y x1 x2 i.a) (treat), nneighbor(4)
```

## Menu

Statistics > Treatment effects > Continuous outcomes > Nearest-neighbor matching  
Statistics > Treatment effects > Binary outcomes > Nearest-neighbor matching  
Statistics > Treatment effects > Count outcomes > Nearest-neighbor matching  
Statistics > Treatment effects > Fractional outcomes > Nearest-neighbor matching  
Statistics > Treatment effects > Nonnegative outcomes > Nearest-neighbor matching

## Syntax

```
teffects nnmatch (ovar ovarlist) (tvar) [if] [in] [weight]
[ , stat options ]
```

*ovar* is a binary, count, continuous, fractional, or nonnegative outcome of interest.

*ovarlist* specifies the covariates in the outcome model.

*tvar* must contain integer values representing the treatment levels. Only two treatment levels are allowed.

---

<i>stat</i>	Description
-------------	-------------

---

### Stat

<code>ate</code>	estimate average treatment effect in population; the default
<code>atet</code>	estimate average treatment effect on the treated

---



---

<i>options</i>	Description
----------------	-------------

---

### Model

<code>nneighbor(#)</code>	specify number of matches per observation; default is <code>nneighbor(1)</code>
<code>biasadj(varlist)</code>	correct for large-sample bias using specified variables
<code>ematch(varlist)</code>	match exactly on specified variables

### SE/Robust

<code>vce(vctype)</code>	<i>vctype</i> may be <ul style="list-style-type: none"> <li><code>vce(robust [ , nn(#) ])</code>; use robust Abadie–Imbens standard errors with # matches</li> <li><code>vce(iid)</code>; use independently and identically distributed Abadie–Imbens standard errors</li> </ul>
--------------------------	--

### Reporting

<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>dmvariables</code>	display names of matching variables
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

### Advanced

<code>caliper(#)</code>	specify the maximum distance for which two observations are potential neighbors
<code>dtolerance(#)</code>	set maximum distance between individuals considered equal
<code>osample(newvar)</code>	<i>newvar</i> identifies observations that violate the overlap assumption
<code>control(#   label)</code>	specify the level of <i>tvar</i> that is the control
<code>tlevel(#   label)</code>	specify the level of <i>tvar</i> that is the treatment
<code>generate(stub)</code>	generate variables containing the observation numbers of the nearest neighbors
<code>metric(metric)</code>	select distance metric for covariates
<code>coeflegend</code>	display legend instead of statistics

---

<i>metric</i>	Description
<u>m</u> ahalanobis	inverse sample covariate covariance; the default
<u>i</u> variance	inverse diagonal sample covariate covariance
<u>e</u> uclidean	identity
<u>m</u> atrix <i>matname</i>	user-supplied scaling matrix

*omvarlist* may contain factor variables; see [U] 11.4.3 Factor variables.

*by*, *collect*, and *statsby* are allowed; see [U] 11.1.10 Prefix commands.

*fweights* are allowed; see [U] 11.1.6 weight.

*coeflegend* does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

## Options

### Model

**nneighbor**(*#*) specifies the number of matches per observation. The default is **nneighbor**(1). Each observation is matched with at least the specified number of observations from the other treatment level. **nneighbor**() must specify an integer greater than or equal to 1 but no larger than the number of observations in the smallest treatment group.

**biasadj**(*varlist*) specifies that a linear function of the specified covariates be used to correct for a large-sample bias that exists when matching on more than one continuous covariate. By default, no correction is performed.

Abadie and Imbens (2006, 2011) show that nearest-neighbor matching estimators are not consistent when matching on two or more continuous covariates and propose a bias-corrected estimator that is consistent. The correction term uses a linear function of variables specified in **biasadj**(); see example 3.

**ematch**(*varlist*) specifies that the variables in *varlist* match exactly. All variables in *varlist* must be numeric and may be specified as factors. **teffects nnmatch** exits with an error if any observations do not have the requested exact match.

### Stat

*stat* is one of two statistics: **ate** or **atet**. **ate** is the default.

**ate** specifies that the average treatment effect be estimated.

**atet** specifies that the average treatment effect on the treated be estimated.

### SE/Robust

**vce**(*vcetype*) specifies the standard errors that are reported. By default, **teffects nnmatch** uses robust standard errors estimated using two matches.

**vce**(robust [*, nn*(*#*)]) specifies that robust standard errors be reported and that the requested number of matches be used optionally.

**vce**(iid) specifies that standard errors for independently and identically distributed data be reported.

The standard derivative-based standard-error estimators cannot be used by **teffects nnmatch**, because these matching estimators are not differentiable. The implemented methods were derived by Abadie and Imbens (2006, 2011, 2012); see *Methods and formulas*.



As discussed in [Abadie and Imbens \(2008\)](#), bootstrap estimators do not provide reliable standard errors for the estimator implemented by `teffects nnmatch`.

## Reporting

`level(#)`; see [\[R\] Estimation options](#).

`dmvariables` specifies that the matching variables be displayed.

`display_options`: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fwrap(#)`, `fwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [\[R\] Estimation options](#).

## Advanced

`caliper(#)` specifies the maximum distance at which two observations are a potential match. By default, all observations are potential matches regardless of how dissimilar they are.

The distance is based on `omvarlist`. If an observation does not have at least `nneighbor(#)` matches, `teffects nnmatch` exits with an error message. Use option `osample(newvar)` to identify all observations that are deficient in matches.

`dtolerance(#)` specifies the tolerance used to determine exact matches. The default value is `dtolerance(sqrt(c(epsdouble)))`.

Integer-valued variables are usually used for exact matching. The `dtolerance()` option is useful when continuous variables are used for exact matching.

`osample(newvar)` specifies that indicator variable `newvar` be created to identify observations that violate the overlap assumption. This variable will identify all observations that do not have at least `nneighbor(#)` matches in the opposite treatment group within `caliper(#)` (for `metric()` distance matching) or `dtolerance(#)` (for `ematch(varlist)` exact matches).

The `vce(robust, nn(#))` option also requires at least `#` matches in the same treatment group within the distance specified by `caliper(#)` or within the exact matches specified by `dtolerance(#)`.

The average treatment effect on the treated, option `atet`, using `vce(iid)` requires only `nneighbor(#)` control group matches for the treated group.

`control(# | label)` specifies the level of `tvar` that is the control. The default is the first treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(# | label)` specifies the level of `tvar` that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

`generate(stub)` specifies that the observation numbers of the nearest neighbors be stored in the new variables `stub1`, `stub2`, ... This option is required if you wish to perform postestimation based on the matching results. The number of variables generated may be more than `nneighbor(#)` because of tied distances. These variables may not already exist.

`metric(metric)` specifies the distance matrix used as the weight matrix in a quadratic form that transforms the multiple distances into a single distance measure; see [Nearest-neighbor matching estimator](#) in *Methods and formulas* for details.

The following option is available with `teffects nnmatch` but is not shown in the dialog box:

`coeflegend`; see [\[R\] Estimation options](#).

## Remarks and examples

The NNM method of treatment-effect estimation imputes the missing potential outcome for each individual by using an average of the outcomes of similar subjects that receive the other treatment level. Similarity between subjects is based on a weighted function of the covariates for each observation. The average treatment effect (ATE) is computed by taking the average of the difference between the observed and potential outcomes for each subject.

`teffects nnmatch` determines the “nearest” by using a weighted function of the covariates for each observation. By default, the Mahalanobis distance is used, in which the weights are based on the inverse of the covariates’ variance–covariance matrix. `teffects nnmatch` also allows you to request exact matching for categorical covariates. For example, you may want to force all matches to be of the same gender or race.

NNM is nonparametric in that no explicit functional form for either the outcome model or the treatment model is specified. This flexibility comes at a price; the estimator needs more data to get to the true value than an estimator that imposes a functional form. More formally, the NNM estimator converges to the true value at a rate slower than the parametric rate, which is the square root of the sample size, when matching on more than one continuous covariate. `teffects nnmatch` uses bias correction to fix this problem. `teffects psmatch` implements an alternative to bias correction; the method matches on a single continuous covariate, the estimated treatment probabilities. See [\[TE\] teffects intro](#) or [\[TE\] teffects intro advanced](#) for more information about this estimator.

We will illustrate the use of `teffects nnmatch` by using data from a study of the effect of a mother’s smoking status during pregnancy (`mbsmoke`) on infant birthweight (`bweight`) as reported by [Cattaneo \(2010\)](#). This dataset also contains information about each mother’s age (`mage`), education level (`medu`), marital status (`mmarried`), whether the first prenatal exam occurred in the first trimester (`prenatal1`), whether this baby was the mother’s first birth (`fbaby`), and the father’s age (`fage`).

### ► Example 1: Estimating the ATE

We begin by using `teffects nnmatch` to estimate the average treatment effect of `mbsmoke` on `bweight`. Subjects are matched using the Mahalanobis distance defined by covariates `mage`, `prenatal1`, `mmarried`, and `fbaby`.

```
. use https://www.stata-press.com/data/r17/cattaneo3
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects nnmatch (bweight mage prenatal1 mmarried fbaby) (mbsmoke)

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : nearest-neighbor matching      Matches: requested =      1
Outcome model  : matching                        min =              1
Distance metric: Mahalanobis                    max =             139
```

	Coefficient	AI robust std. err.	z	P> z	[95% conf. interval]	
ATE						
mbsmoke (smoker vs nonsmoker)	-240.3306	28.43006	-8.45	0.000	-296.0525	-184.6087

The average birthweight if all mothers were to smoke would be 240 grams less than the average that would occur if none of the mothers had smoked.

When the model includes indicator and categorical variables, you may want to restrict matches to only those subjects who are in the same category. The `ematch()` option of `teffects nnmatch` allows you to specify such variables that must match exactly.

### ▷ Example 2: Exact matching

Here we use the `ematch()` option to require exact matches on the binary variables `prenatal1`, `mmarried`, and `fbaby`. We also use Euclidean distance, rather than the default Mahalanobis distance, to match on the continuous variable `mage`.

```
. teffects nnmatch (bweight mage) (mbsmoke),
> ematch(prenatal1 mmarrried fbaby) metric(euclidean)
Treatment-effects estimation      Number of obs      =      4,642
Estimator      : nearest-neighbor matching      Matches: requested =      1
Outcome model  : matching                      min =      1
Distance metric: Euclidean                    max =      139
```

	bweight	Coefficient	AI robust std. err.	z	P> z	[95% conf. interval]
ATE	mbsmoke (smoker vs nonsmoker)	-240.3306	28.43006	-8.45	0.000	-296.0525 -184.6087

◀

Abadie and Imbens (2006, 2011) have shown that nearest-neighbor matching estimators are not consistent when matching on two or more continuous covariates. A bias-corrected estimator that uses a linear function of variables can be specified with `biasadj()`.

### ▷ Example 3: Bias adjustment

Here we match on two continuous variables, `mage` and `fage`, and we use the bias-adjusted estimator:

```
. teffects nnmatch (bweight mage fage) (mbsmoke),
> ematch(prenatal1 mmarrried fbaby) biasadj(mage fage)
Treatment-effects estimation      Number of obs      =      4,642
Estimator      : nearest-neighbor matching      Matches: requested =      1
Outcome model  : matching                      min =      1
Distance metric: Mahalanobis                    max =      25
```

	bweight	Coefficient	AI robust std. err.	z	P> z	[95% conf. interval]
ATE	mbsmoke (smoker vs nonsmoker)	-223.8389	26.19973	-8.54	0.000	-275.1894 -172.4883

These results are similar to those reported in [example 1](#).

◀

## ▷ Example 4: NNM can reduce to RA

NNM reduces to RA when matching exactly and all the covariates are discrete. We begin our illustration of this point by estimating the ATE by NNM using exact matching on `mmarried` and the mother's age-categories `magecat`.

```
. teffects nmatch (bweight) (mbsmoke), ematch(i.mmarried i.magecat)
Treatment-effects estimation           Number of obs   =      4,642
Estimator       : nearest-neighbor matching   Matches: requested =      1
Outcome model   : matching                   min           =     11
Distance metric: Mahalanobis                 max           =    1310
```

bweight	Coefficient	AI robust std. err.	z	P> z	[95% conf. interval]	
ATE						
mbsmoke (smoker vs nonsmoker)	-241.5264	24.39661	-9.90	0.000	-289.3429	-193.71

The RA estimator that includes the interactions among the discrete covariates produces the same point estimate.

```
. teffects ra (bweight i.mmarried##i.magecat) (mbsmoke)
Iteration 0:   EE criterion = 1.523e-23
Iteration 1:   EE criterion = 7.899e-26
Treatment-effects estimation           Number of obs   =      4,642
Estimator       : regression adjustment
Outcome model   : linear
Treatment model: none
```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
mbsmoke (smoker vs nonsmoker)	-241.5264	24.26233	-9.95	0.000	-289.0797	-193.9732
POmean						
mbsmoke nonsmoker	3403.651	9.492683	358.56	0.000	3385.046	3422.256

The two estimates of the ATE are the same. The standard errors differ in finite samples because the RA and NNM estimators use different robust estimators of the variance of the estimator.

With exact matching on discrete covariates, the NNM estimator reduces to an average of differences in cell means. With fully interacted discrete covariates, the RA estimator reduces to the same average of difference in cell means.

## Video example

Treatment effects in Stata: Nearest-neighbor matching

## Stored results

teffects nnmatch stores the following in `e()`:

### Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level $j$
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(k_nnneighbor)</code>	requested number of matches
<code>e(k_nnmin)</code>	minimum number of matches
<code>e(k_nnmax)</code>	maximum number of matches
<code>e(k_robust)</code>	matches for robust VCE

### Macros

<code>e(cmd)</code>	teffects
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(emvarlist)</code>	exact match variables
<code>e(bavarlist)</code>	variables used in bias adjustment
<code>e(mvarlist)</code>	match variables
<code>e(subcmd)</code>	nnmatch
<code>e(metric)</code>	mahalanobis, ivariance, euclidean, or matrix <i>matname</i>
<code>e(stat)</code>	statistic estimated, ate or atet
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vctype)</code>	title used to label Std. err.
<code>e(datasignature)</code>	the checksum
<code>e(datasignaturevars)</code>	variables used in calculation of checksum
<code>e(properties)</code>	$b$ $V$
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

### Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

### Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

### Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, $p$ -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r`-class command is run after the estimation command.

## Methods and formulas

The methods and formulas presented here provide the technical details underlying the estimators implemented in `teffects nmatch` and `teffects psmatch`. See *Methods and formulas* of [TE] **teffects aipw** for the methods and formulas used by `teffects aipw`, `teffects ipw`, `teffects ipwra`, and `teffects ra`.

Methods and formulas are presented under the following headings:

*Nearest-neighbor matching estimator*  
*Bias-corrected matching estimator*  
*Propensity-score matching estimator*  
*PSM, ATE, and ATET variance adjustment*

### Nearest-neighbor matching estimator

`teffects nmatch` implements the nearest-neighbor matching (NNM) estimator for the average treatment effect (ATE) and the average treatment effect on the treated (ATET). This estimator was derived by Abadie and Imbens (2006, 2011) and was previously implemented in Stata as discussed in Abadie et al. (2004).

`teffects psmatch` implements nearest-neighbor matching on an estimated propensity score. A propensity score is a conditional probability of treatment. The standard errors implemented in `teffects psmatch` were derived by Abadie and Imbens (2012).

`teffects nmatch` and `teffects psmatch` permit two treatment levels: the treatment group with  $t = 1$  and a control group with  $t = 0$ .

Matching estimators are based on the potential-outcome model, in which each individual has a well-defined outcome for each treatment level; see [TE] **teffects intro**. In the binary-treatment potential-outcome model,  $y_1$  is the potential outcome obtained by an individual if given treatment-level 1 and  $y_0$  is the potential outcome obtained by each individual  $i$  if given treatment-level 0. The problem posed by the potential-outcome model is that only  $y_{1i}$  or  $y_{0i}$  is observed, never both.  $y_{0i}$  and  $y_{1i}$  are realizations of the random variables  $y_0$  and  $y_1$ . Throughout this document,  $i$  subscripts denote realizations of the corresponding, unsubscripted random variables.

Formally, the ATE is

$$\tau_1 = E(y_1 - y_0)$$

and the ATET is

$$\delta_1 = E(y_1 - y_0 | t = 1)$$

These expressions imply that we must have some solution to the missing-data problem that arises because we only observe either  $y_{1i}$  or  $y_{0i}$ , not both.

For each individual, NNM uses an average of the individuals that are most similar, but get the other treatment level, to predict the unobserved potential outcome. NNM uses the covariates  $\{x_1, x_2, \dots, x_p\}$  to find the most similar individuals that get the other treatment level.

More formally, consider the vector of covariates  $\mathbf{x}_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,p}\}$  and frequency weight  $w_i$  for observation  $i$ . The distance between  $\mathbf{x}_i$  and  $\mathbf{x}_j$  is parameterized by the vector norm

$$\|\mathbf{x}_i - \mathbf{x}_j\|_S = \{(\mathbf{x}_i - \mathbf{x}_j)' \mathbf{S}^{-1} (\mathbf{x}_i - \mathbf{x}_j)\}^{1/2}$$

where  $\mathbf{S}$  is a given symmetric, positive-definite matrix.

Using this distance definition, we find that the set of nearest-neighbor indices for observation  $i$  is

$$\Omega_m^{\mathbf{x}}(i) = \{j_1, j_2, \dots, j_{m_i} \mid t_{j_k} = 1 - t_i, \|\mathbf{x}_i - \mathbf{x}_{j_k}\|_S < \|\mathbf{x}_i - \mathbf{x}_l\|_S, t_l = 1 - t_i, l \neq j_k\}$$

Here  $m_i$  is the smallest number such that the number of elements in each set,  $m_i = |\Omega_m^x(i)| = \sum_{j \in \Omega_m^x(i)} w_j$ , is at least  $m$ , the desired number of matches. You set the size of  $m$  using the `mneighbor(#)` option. The number of matches for the  $i$ th observation may not equal  $m$  because of ties or if there are not enough observations with a distance from observation  $i$  within the caliper limit,  $c$ ,  $\|\mathbf{x}_i - \mathbf{x}_j\|_S \leq c$ . You may set the caliper limit by using the `caliper(#)` option. For ease of notation, we will use the abbreviation  $\Omega(i) = \Omega_m^x(i)$ .

With the `metric(string)` option, you have three choices for the scaling matrix  $\mathbf{S}$ : Mahalanobis, inverse variance, or Euclidean.

$$\mathbf{S} = \begin{cases} \frac{(\mathbf{X} - \bar{\mathbf{x}}' \mathbf{1}_n)' \mathbf{W} (\mathbf{X} - \bar{\mathbf{x}}' \mathbf{1}_n)}{\sum_i w_i - 1} & \text{if metric = mahalanobis} \\ \text{diag} \left\{ \frac{(\mathbf{X} - \bar{\mathbf{x}}' \mathbf{1}_n)' \mathbf{W} (\mathbf{X} - \bar{\mathbf{x}}' \mathbf{1}_n)}{\sum_i w_i - 1} \right\} & \text{if metric = ivariance} \\ \mathbf{I}_p & \text{if metric = euclidean} \end{cases}$$

where  $\mathbf{1}_n$  is an  $n \times 1$  vector of ones,  $\mathbf{I}_p$  is the identity matrix of order  $p$ ,  $\bar{\mathbf{x}} = (\sum_i w_i \mathbf{x}_i) / (\sum_i w_i)$ , and  $\mathbf{W}$  is an  $n \times n$  diagonal matrix containing frequency weights.

The NNM method predicting the potential outcome for the  $i$ th observation as a function of the observed  $y_i$  is

$$\hat{y}_{ti} = \begin{cases} y_i & \text{if } t_i = t \\ \frac{\sum_{j \in \Omega(i)} w_j y_j}{\sum_{j \in \Omega(i)} w_j} & \text{otherwise} \end{cases}$$

for  $t \in \{0, 1\}$ .

We are now set to provide formulas for estimates  $\hat{\tau}_1$ , the ATE, and  $\hat{\delta}_1$ , the ATET,

$$\hat{\tau}_1 = \frac{\sum_{i=1}^n w_i (\hat{y}_{1i} - \hat{y}_{0i})}{\sum_{i=1}^n w_i} = \frac{\sum_{i=1}^n w_i (2t_i - 1) \{1 + K_m(i)\} y_i}{\sum_{i=1}^n w_i}$$

$$\hat{\delta}_1 = \frac{\sum_{i=1}^n t_i w_i (\hat{y}_{1i} - \hat{y}_{0i})}{\sum_{i=1}^n t_i w_i} = \frac{\sum_{i=1}^n \{t_i - (1 - t_i) K_m(i)\} y_i}{\sum_{i=1}^n t_i w_i}$$

where

$$K_m(i) = \sum_{j=1}^n I\{i \in \Omega(j)\} \frac{w_j}{\sum_{k \in \Omega(j)} w_k}$$

The estimated variance of  $\hat{\tau}_1$  and  $\hat{\delta}_1$  are computed as

$$\hat{\sigma}_\tau^2 = \frac{\sum_{i=1}^n w_i \left[ (\hat{y}_{1i} - \hat{y}_{0i} - \hat{\tau}_1)^2 + \hat{\xi}_i^2 \{K_m^2(i) + 2K_m(i) - K'_m(i)\} \right]}{(\sum_{i=1}^n w_i)^2}$$

$$\hat{\sigma}_\delta^2 = \frac{\sum_{i=1}^n t_i w_i \left[ (\hat{y}_{1i} - \hat{y}_{0i} - \hat{\delta}_1)^2 + \hat{\xi}_i^2 \{K_m^2(i) - K'_m(i)\} \right]}{(\sum_{i=1}^n t_i w_i)^2}$$

where

$$K'_m(i) = \sum_{j=1}^n I\{i \in \Omega(j)\} \frac{w_j}{\left(\sum_{k \in \Omega(j)} w_k\right)^2}$$

and  $\xi_i^2 = \text{var}(y_{ti} | \mathbf{x}_i)$  is the conditional outcome variance. If we can assume that  $\xi_i^2$  does not vary with the covariates or treatment (homoskedastic), then we can compute an ATE estimate of  $\xi_\tau^2$  as

$$\widehat{\xi}_\tau^2 = \frac{1}{2 \sum_i^n w_i} \sum_{i=1}^n w_i \left[ \frac{\sum_{j \in \Omega(i)} w_j \{y_i - y_j(1 - t_i) - \widehat{\tau}_1\}^2}{\sum_{j \in \Omega(i)} w_j} \right]$$

and an ATET estimate of  $\xi_\delta^2$  as

$$\widehat{\xi}_\delta^2 = \frac{1}{2 \sum_i^n t_i w_i} \sum_{i=1}^n t_i w_i \left[ \frac{\sum_{j \in \Omega(i)} t_j w_j \{y_i - y_j(1 - t_i) - \widehat{\delta}_1\}^2}{\sum_{j \in \Omega(i)} t_j w_j} \right]$$

If the conditional outcome variance is dependent on the covariates or treatment, we require an estimate for  $\xi_i^2$  at each observation. In this case, we require a second matching procedure, where we match on observations within the same treatment group.

Define the within-treatment matching set

$$\Psi_h^{\mathbf{x}}(i) = \{j_1, j_2, \dots, j_{h_i} \mid t_{j_k} = t_i, \|\mathbf{x}_i - \mathbf{x}_{j_k}\|_S < \|\mathbf{x}_i - \mathbf{x}_l\|_S, t_l = t_i, l \neq j_k\}$$

where  $h$  is the desired set size. As before, the number of elements in each set,  $h_i = |\Psi_h^{\mathbf{x}}(i)|$ , may vary depending on ties and the value of the caliper. You set  $h$  using the `vce(robust, nn(#))` option. As before, we will use the abbreviation  $\Psi(i) = \Psi_h^{\mathbf{x}}(i)$  where convenient.

We estimate  $\xi_i^2$  by

$$\widehat{\xi}_{t_i}^2(\mathbf{x}_i) = \frac{\sum_{j \in \Psi(i)} w_j (y_j - \bar{y}_{\Psi(i)})^2}{\sum_{j \in \Psi(i)} w_j - 1} \quad \text{where} \quad \bar{y}_{\Psi(i)} = \frac{\sum_{j \in \Psi(i)} w_j y_j}{\sum_{j \in \Psi(i)} w_j - 1}$$

### Bias-corrected matching estimator

When matching on more than one continuous covariate, the matching estimator described above is biased, even in infinitely large samples; in other words, it is not  $\sqrt{n}$ -consistent; see [Abadie and Imbens \(2006, 2011\)](#). Following [Abadie and Imbens \(2011\)](#) and [Abadie et al. \(2004\)](#), `teffects nmatch` makes an adjustment based on the regression functions  $\mu_t(\tilde{\mathbf{x}}_i) = E(y_t \mid \tilde{\mathbf{x}} = \tilde{\mathbf{x}}_i)$ , for  $t = 0, 1$  and the set of covariates  $\tilde{\mathbf{x}}_i = (\tilde{x}_{i,1}, \dots, \tilde{x}_{i,q})$ . The bias-correction covariates may be the same as the NNM covariates  $\mathbf{x}_i$ . We denote the least-squares estimates as  $\widehat{\mu}_t(\tilde{\mathbf{x}}_i) = \widehat{v}_t + \widehat{\beta}'_t \tilde{\mathbf{x}}_i$ , where we regress  $\{y_i \mid t_i = t\}$  onto  $\{\tilde{\mathbf{x}}_i \mid t_i = t\}$  with weights  $w_i K_m(i)$ , for  $t = 0, 1$ .



Given the estimated regression functions, the bias-corrected predictions for the potential outcomes are computed as

$$\hat{y}_{ti} = \begin{cases} y_i & \text{if } t_i = t \\ \frac{\sum_{j \in \Omega_m^{\mathbf{x}}(i)} w_j \{y_j + \hat{\mu}_t(\tilde{\mathbf{x}}_i) - \hat{\mu}_t(\tilde{\mathbf{x}}_j)\}}{\sum_{j \in \Omega_m^{\mathbf{x}}(i)} w_j} & \text{otherwise} \end{cases}$$

The `biasadj` (*varlist*) option specifies the bias-adjustment covariates  $\tilde{\mathbf{x}}_i$ .

## Propensity-score matching estimator

The propensity-score matching (PSM) estimator uses a treatment model (TM),  $p(\mathbf{z}_i, t, \gamma)$ , to model the conditional probability that observation  $i$  receives treatment  $t$  given covariates  $\mathbf{z}_i$ . The literature calls  $p(\mathbf{z}_i, t, \gamma)$  a propensity score, and PSM matches on the estimated propensity scores.

When matching on the estimated propensity score, the set of nearest-neighbor indices for observation  $i$ ,  $i = 1, \dots, n$ , is

$$\Omega_m^p(i) = \{j_1, j_2, \dots, j_{m_i} \mid t_{j_k} = 1 - t_i, |\hat{p}_i(t) - \hat{p}_{j_k}(t)| < |\hat{p}_i(t) - \hat{p}_l(t)|, t_l = 1 - t_i, l \neq j_k\}$$

where  $\hat{p}_i(t) = p(\mathbf{z}_i, t, \hat{\gamma})$ . As was the case with the NNM estimator,  $m_i$  is the smallest number such that the number of elements in each set,  $m_i = |\Omega_m^p(i)| = \sum_{j \in \Omega_m^p(i)} w_j$ , is at least  $m$ , the desired number of matches, set by the `nneighbor(#)` option.

We define the within-treatment matching set analogously,

$$\Psi_h^p(i) = \{j_1, j_2, \dots, j_{h_i} \mid t_{j_k} = t_i, |\hat{p}_i(t) - \hat{p}_{j_k}(t)| < |\hat{p}_i(t) - \hat{p}_l(t)|, t_l = t_i, l \neq j_k\}$$

where  $h$  is the desired number of within-treatment matches, and  $h_i = |\Psi_h^p(i)|$ , for  $i = 1, \dots, n$ , may vary depending on ties and the value of the caliper. The sets  $\Psi_h^p(i)$  are required to compute standard errors for  $\hat{\tau}_1$  and  $\hat{\delta}_1$ .

Once a matching set is computed for each observation, the potential-outcome mean, ATE, and ATET computations are identical to those of NNM. The ATE and ATET standard errors, however, must be adjusted because the TM parameters were estimated; see [Abadie and Imbens \(2012\)](#).

## PSM, ATE, and ATET variance adjustment

The variances for  $\hat{\tau}_1$  and  $\hat{\delta}_1$  must be adjusted because we use  $\hat{\gamma}$  instead of  $\gamma$ . The adjusted variances for  $\hat{\tau}_1$  and  $\hat{\delta}_1$  have the following forms, respectively:

$$\begin{aligned} \hat{\sigma}_{\tau, \text{adj}}^2 &= \hat{\sigma}_{\tau}^2 + \hat{\mathbf{c}}_{\tau}' \hat{\mathbf{V}}_{\gamma} \hat{\mathbf{c}}_{\tau} \\ \hat{\sigma}_{\delta, \text{adj}}^2 &= \hat{\sigma}_{\delta}^2 - \hat{\mathbf{c}}_{\delta}' \hat{\mathbf{V}}_{\gamma} \hat{\mathbf{c}}_{\delta} + \frac{\partial \hat{\delta}_1}{\partial \gamma'} \hat{\mathbf{V}}_{\gamma} \frac{\partial \hat{\delta}_1}{\partial \gamma} \end{aligned}$$

In both equations, the matrix  $\hat{\mathbf{V}}_{\gamma}$  is the TM coefficient variance–covariance matrix.

The adjustment term for ATE can be expressed as

$$\widehat{\mathbf{c}}_\tau = \frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^n w_i f(\mathbf{z}'_i \widehat{\boldsymbol{\gamma}}) \left( \frac{\widehat{\text{cov}}(\mathbf{z}_i, \widehat{y}_{i1})}{\widehat{p}_i(1)} + \frac{\widehat{\text{cov}}(\mathbf{z}_i, \widehat{y}_{i0})}{\widehat{p}_i(0)} \right)$$

where

$$f(\mathbf{z}'_i \widehat{\boldsymbol{\gamma}}) = \frac{d p(\mathbf{z}_i, 1, \widehat{\boldsymbol{\gamma}})}{d(\mathbf{z}'_i \widehat{\boldsymbol{\gamma}})}$$

is the derivative of  $p(\mathbf{z}_i, 1, \widehat{\boldsymbol{\gamma}})$  with respect to  $\mathbf{z}'_i \widehat{\boldsymbol{\gamma}}$ , and

$$\widehat{\text{cov}}(\mathbf{z}_i, \widehat{y}_{ti}) = \begin{cases} \frac{\sum_{j \in \Psi_h(i)} w_j (\mathbf{z}_j - \bar{\mathbf{z}}_{\Psi_i})(y_j - \bar{y}_{\Psi_i})}{\sum_{j \in \Psi_h(i)} w_j - 1} & \text{if } t_i = t \\ \frac{\sum_{j \in \Omega_h(i)} w_j (\mathbf{z}_j - \bar{\mathbf{z}}_{\Omega_i})(y_j - \bar{y}_{\Omega_i})}{\sum_{j \in \Omega_h(i)} w_j - 1} & \text{otherwise} \end{cases}$$

is a  $p \times 1$  vector with

$$\bar{\mathbf{z}}_{\Psi_i} = \frac{\sum_{j \in \Psi_h(i)} w_j \mathbf{z}_j}{\sum_{j \in \Psi_h(i)} w_j} \quad \bar{\mathbf{z}}_{\Omega_i} = \frac{\sum_{j \in \Omega_h(i)} w_j \mathbf{z}_j}{\sum_{j \in \Omega_h(i)} w_j} \quad \text{and} \quad \bar{y}_{\Omega_i} = \frac{\sum_{j \in \Omega_h(i)} w_j y_j}{\sum_{j \in \Omega_h(i)} w_j}$$

Here we have used the notation  $\Psi_h(i) = \Psi_h^p(i)$  and  $\Omega_h(i) = \Omega_h^p(i)$  to stress that the within-treatment and opposite-treatment clusters used in computing  $\widehat{\sigma}_{\tau, \text{adj}}^2$  and  $\widehat{\delta}_{\tau, \text{adj}}^2$  are based on  $h$  instead of the cluster  $\Omega_m^p(i)$  based on  $m$  used to compute  $\widehat{\tau}_1$  and  $\widehat{\delta}_1$ , although you may desire to have  $h = m$ .

The adjustment term  $\mathbf{c}_\delta$  for the ATET estimate has two components,  $\mathbf{c}_\delta = \mathbf{c}_{\delta,1} + \mathbf{c}_{\delta,2}$ , defined as

$$\mathbf{c}_{\delta,1} = \frac{1}{\sum_{i=1}^n t_i w_i} \sum_{i=1}^n w_i \mathbf{z}_i f(\mathbf{z}'_i \widehat{\boldsymbol{\gamma}}) (\widehat{y}_{1i} - \widehat{y}_{0i} - \widehat{\delta}_1)$$

$$\mathbf{c}_{\delta,2} = \frac{1}{\sum_{i=1}^n t_i w_i} \sum_{i=1}^n w_i f(\mathbf{z}'_i \widehat{\boldsymbol{\gamma}}) \left\{ \widehat{\text{cov}}(\mathbf{z}_i, \widehat{y}_{1i}) + \frac{\widehat{p}_i(1)}{\widehat{p}_i(0)} \widehat{\text{cov}}(\mathbf{z}_i, \widehat{y}_{0i}) \right\}$$

where

$$\widetilde{y}_{ti} = \begin{cases} \frac{\sum_{j \in \Psi_h(-i)} w_j y_j}{\sum_{j \in \Psi_h(-i)} w_j} & \text{if } t = t_i \\ \frac{\sum_{j \in \Omega_h} w_j y_j}{\sum_{j \in \Omega_h} w_j} & \text{otherwise} \end{cases}$$

and the within-treatment matching sets  $\Psi_h(-i) = \Psi_h^p(-i)$  are similar to  $\Psi_h^p(i)$  but exclude observation  $i$ :

$$\Psi_h^p(-i) = \{j_1, j_2, \dots, j_{h_i} \mid j_k \neq i, t_{j_k} = t_i, |\widehat{p}_j - \widehat{p}_{j_k}| < |\widehat{p}_i - \widehat{p}_l|, t_l = t_i, l \notin \{i, j_k\}\}$$

Finally, we cover the computation of  $\widehat{\frac{\partial \delta_1}{\partial \gamma}}$  in the third term on the right-hand side of  $\widehat{\sigma}_{\delta, \text{adj}}^2$ . Here we require yet another clustering, but we match on the opposite treatment by using the covariates  $\mathbf{z}_i = (z_{i,1}, \dots, z_{i,p})'$ . We will denote these cluster sets as  $\Omega_m^{\mathbf{z}}(i)$ , for  $i = 1, \dots, n$ .

The estimator of the  $p \times 1$  vector  $(\partial \delta_1) / (\partial \gamma)$  is computed as

$$\widehat{\frac{\partial \delta_1}{\partial \gamma}} = \frac{1}{\sum_i^n t_i w_i} \sum_{i=1}^n \mathbf{z}_i f(\mathbf{z}' \hat{\gamma}) \left\{ (2t_i - 1)(y_i - \bar{y}_{\Omega_m^{\mathbf{z}}(i)}) - \hat{\delta}_1 \right\}$$

where

$$\bar{y}_{\Omega_m^{\mathbf{z}}(i)} = \frac{\sum_{j \in \Omega_m^{\mathbf{z}}(i)} w_j y_j}{\sum_{j \in \Omega_m^{\mathbf{z}}(i)} w_j}$$

## References

- Abadie, A., D. M. Drukker, J. L. Herr, and G. W. Imbens. 2004. Implementing matching estimators for average treatment effects in Stata. *Stata Journal* 4: 290–311.
- Abadie, A., and G. W. Imbens. 2006. Large sample properties of matching estimators for average treatment effects. *Econometrica* 74: 235–267. <https://doi.org/10.1111/j.1468-0262.2006.00655.x>.
- . 2008. On the failure of the bootstrap for matching estimators. *Econometrica* 76: 1537–1557. <https://doi.org/10.3982/ECTA6474>.
- . 2011. Bias-corrected matching estimators for average treatment effects. *Journal of Business and Economic Statistics* 29: 1–11. <https://doi.org/10.1198/jbes.2009.07333>.
- . 2012. Matching on the estimated propensity score. Harvard University and National Bureau of Economic Research. <http://www.nber.org/papers/w15301>.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Díaz, J. D., I. Gutiérrez, and J. Rivera. 2021. Implementing blopmatching in Stata. *Stata Journal* 21: 180–194.
- Drukker, D. M. 2016. Exact matching on discrete covariates is the same as regression adjustment. *The Stata Blog: Not Elsewhere Classified*. <http://blog.stata.com/2016/08/16/exact-matching-on-discrete-covariates-is-the-same-as-regression-adjustment/>.
- Huber, C. 2015. Introduction to treatment effects in Stata: Part 2. *The Stata Blog: Not Elsewhere Classified*. <http://blog.stata.com/2015/08/24/introduction-to-treatment-effects-in-stata-part-2/>.

## Also see

- [TE] [teffects postestimation](#) — Postestimation tools for teffects
- [TE] [teffects](#) — Treatment-effects estimation for observational data
- [TE] [teffects psmatch](#) — Propensity-score matching
- [U] [20 Estimation and postestimation commands](#)

[Postestimation commands](#)[predict](#)[Remarks and examples](#)[Also see](#)

## Postestimation commands

The following postestimation command is of special interest after `teffects`:

Command	Description
<code>teoverlap</code>	overlap plots
<code>tebalance</code>	check balance of covariates

The following standard postestimation commands are also available:

Command	Description
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
<code>hausman</code>	Hausman’s specification test
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of coefficients
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
<code>predict</code>	treatment effects, conditional means at treatment, propensity scores, etc.
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

# predict

## Description for predict

`predict` creates a new variable containing predictions such as treatment effects, potential outcomes, conditional means, propensity scores, linear predictions, nearest-neighbor distances, and log square root of latent variances.

## Menu for predict

Statistics > Postestimation

## Syntaxes for predict

Syntaxes are presented under the following headings:

*Syntax for predict after aipw and ipwra*  
*Syntax for predict after ipw*  
*Syntax for predict after nnmatch and psmatch*  
*Syntax for predict after ra*

### Syntax for predict after aipw and ipwra

```
predict [type] { stub* | newvar | newvarlist } [if] [in]
      [, statistic tlevel(treat_level)]
```

```
predict [type] stub* [if] [in], scores
```

<i>statistic</i>	Description
Main	
<code>te</code>	treatment effect; the default
<code>cmean</code>	conditional mean at treatment level
<code>ps</code>	propensity score
<code>xb</code>	linear prediction
<code>psxb</code>	linear prediction for propensity score
<code>lnsigma</code>	log square root of conditional latent variance (for outcome model <code>hetprobit()</code> ) at treatment level
<code>pslnsigma</code>	log square root of latent variance (for treatment model <code>hetprobit()</code> )

If you do not specify `tlevel()` and only specify one new variable, `te` and `psxb` assume `tlevel()` specifies the first noncontrol treatment level.

If you do not specify `tlevel()` and only specify one new variable, `cmean`, `ps`, `xb`, and `lnsigma` assume `tlevel()` specifies the first treatment level.

You specify one or  $t$  new variables with `cmean`, `ps`, `xb`, and `lnsigma`, where  $t$  is the number of treatment levels.

You specify one or  $t-1$  new variables with `te`, `psxb`, and `pslnsigma`.

**Syntax for predict after ipw**

```
predict [type] { stub* | newvar | newvarlist } [if] [in]
      [, statistic tlevel(treat_level)]
```

```
predict [type] stub* [if] [in], scores
```

<i>statistic</i>	Description
------------------	-------------

Main

<code>ps</code>	propensity score; the default
<code>xb</code>	linear prediction for the propensity score
<code>lnsigma</code>	log square root of latent variance (for treatment model <code>hetprobit()</code> )

If you do not specify `tlevel()` and only specify one new variable, `ps` assumes `tlevel()` specifies the first treatment level.

If you do not specify `tlevel()` and only specify one new variable, `xb` assumes `tlevel()` specifies the first noncontrol treatment level.

You specify one or  $t$  new variables with `ps`, where  $t$  is the number of treatment levels.

You specify one or  $t-1$  new variables with `xb` and `lnsigma`.

**Syntax for predict after nnmatch and psmatch**

```
predict [type] { stub* | newvarlist } [, statistic tlevel(treat_level)]
```

<i>statistic</i>	Description
------------------	-------------

Main

<code>te</code>	treatment effect; the default
<code>po</code>	potential outcome
<code>distance</code>	nearest-neighbor distance
<code>ps</code>	propensity score ( <code>psmatch</code> only)
<code>lnsigma</code>	log square root of latent variance (for treatment model <code>hetprobit()</code> )

These statistics are available for the estimation sample only and require the estimation option `generate(stub)`. This is because of the nonparametric nature of the matching estimator.

If you do not specify `tlevel()` and only specify one new variable, `po` and `ps` assume `tlevel()` specifies the first treatment level.

You specify one new variable with `te` and `lnsigma`.

You specify one or two new variables with `po` and `ps`.

## Syntax for predict after ra

```
predict [type] { stub* | newvar | newvarlist } [if] [in]
      [, statistic tlevel(treat_level)]
```

```
predict [type] stub* [if] [in], scores
```

<i>statistic</i>	Description
<b>te</b>	treatment effect; the default
<b>cmean</b>	conditional mean at treatment level
<b>xb</b>	linear prediction
<b>lnsigma</b>	log square root of conditional latent variance (for outcome model <code>hetprobit()</code> ) at treatment level

### Main

<b>te</b>	treatment effect; the default
<b>cmean</b>	conditional mean at treatment level
<b>xb</b>	linear prediction
<b>lnsigma</b>	log square root of conditional latent variance (for outcome model <code>hetprobit()</code> ) at treatment level

If you do not specify `tlevel()` and only specify one new variable, `te` assumes `tlevel()` specifies the first noncontrol treatment level.

If you do not specify `tlevel()` and only specify one new variable, `cmean`, `xb`, and `lnsigma` assume `tlevel()` specifies the first treatment level.

You specify one or  $t$  new variables with `cmean`, `xb`, and `lnsigma`, where  $t$  is the number of treatment levels.

You specify one or  $t-1$  new variables with `te`.

## Options for predict

Options are presented under the following headings:

*Options for predict after aipw and ipwra*  
*Options for predict after ipw*  
*Options for predict after nnmatch and psmatch*  
*Options for predict after ra*

## Options for predict after aipw and ipwra

### Main

**te**, the default, calculates the treatment effect for each noncontrol treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

**cmean** calculates the conditional mean for each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

**ps** calculates the propensity score of each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

**xb** calculates the linear prediction at each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`psxb` calculates the linear prediction for the propensity score at each noncontrol level of the treatment or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

`lnsigma` calculates the log square root of the conditional latent variance for each treatment level or the treatment level specified in `tlevel()`. This option is valid when outcome model `hetprobit()` was used. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`pslnsigma` calculates the log square root of the latent variance for the propensity score. This option is only valid when treatment model `hetprobit()` was used. Specify only one new variable.

`tlevel(treat_level)` specifies the treatment level for prediction.

`scores` calculates the score variables. Parameter-level scores are computed for the treatment mean and average treatment-effect equations. Equation-level scores are computed for the propensity-score equations.

The  $j$ th new variable will contain the scores for the  $j$ th parameter in the coefficient table if  $j \leq t$ , where  $t$  is the number of treatment levels. Otherwise, it will contain the scores for fitted equation  $j - t$  following the first  $t$  parameters in the coefficient table.

## Options for predict after ipw

Main

`ps`, the default, calculates the propensity score of each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`xb` calculates the linear prediction for the propensity score at each noncontrol level of the treatment or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

`lnsigma` calculates the log square root of the latent variance. This option is only valid when treatment model `hetprobit()` was used. Specify only one new variable.

`tlevel(treat_level)` specifies the treatment level for prediction.

`scores` calculates the score variables. Parameter-level scores are computed for the treatment mean and average treatment-effect equations. Equation-level scores are computed for the propensity-score equations.

The  $j$ th new variable will contain the scores for the  $j$ th parameter in the coefficient table if  $j \leq t$ , where  $t$  is the number of treatment levels. Otherwise, it will contain the scores for fitted equation  $j - t$  following the first  $t$  parameters in the coefficient table.

## Options for predict after nmatch and psmatch

Main

`te`, the default, calculates the treatment effect.

`po` calculates the predicted potential outcomes for each observation and treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify new variables for the control and treated groups.



`distance` calculates the distances of the nearest neighbors for each observation. The number of variables generated is equal to the maximum number of nearest-neighbor matches. This is equal to the number of index variables generated by the estimation option `generate(stub)`. You may use the `stub*` syntax to set the distance variable prefix: `stub1, stub2, ...`.

`ps` calculates the propensity score of each treatment level or the propensity score of the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify new variables for the control and treated groups.

`lnsigma` calculates the log square root of the latent variance. This option is only valid when treatment model `hetprobit()` was used. Specify only one new variable.

`tlevel(treat_level)` restricts potential-outcome estimation to either the treated group or the control group. This option may only be specified with options `po` and `ps`.

## Options for predict after ra

### Main

`te`, the default, calculates the treatment effect for each noncontrol treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

`cmean` calculates the conditional mean for each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`xb` calculates the linear prediction at each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`lnsigma` calculates the log square root of the conditional latent variance for each treatment level or the treatment level specified in `tlevel()`. This option is valid when outcome model `hetprobit()` was used. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`tlevel(treat_level)` specifies the treatment level for prediction.

`scores` calculates the score variables. Parameter-level scores are computed for the treatment mean and average treatment-effect equations. Equation-level scores are computed for the regression equations.

The  $j$ th new variable will contain the scores for the  $j$ th parameter in the coefficient table if  $j \leq t$ , where  $t$  is the number of treatment levels. Otherwise, it will contain the scores for fitted equation  $j - t$  following the first  $t$  parameters in the coefficient table.

## Remarks and examples

Checking model specification is the most frequent reason for postestimation computation after `teffects`. `teoverlap` provides a graphical method for checking the overlap assumption; see [\[TE\] teoverlap](#). Summarizing the estimated probabilities provides another check. Recall that the reciprocals of these estimated probabilities are used as weights by some of the estimators. If the estimated probabilities are too small, the weights blow up.

We estimate the ATE of maternal smoking on infant birthweight by inverse-probability weighting; see [example 1](#) of [\[TE\] teffects ipw](#) for background.

```
. use https://www.stata-press.com/data/r17/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ipw (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu, probit)
Iteration 0:  EE criterion = 4.621e-21
Iteration 1:  EE criterion = 7.358e-26
Treatment-effects estimation          Number of obs    =    4,642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: probit
```

bweight	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
mbsmoke (Smoker vs Nonsmoker)	-230.6886	25.81524	-8.94	0.000	-281.2856	-180.0917
POmean						
mbsmoke Nonsmoker	3403.463	9.571369	355.59	0.000	3384.703	3422.222

Below we compute and summarize the estimated treatment probabilities.

```
. predict pr1
(option ps assumed; propensity score)
. summarize pr1 if mbsmoke==1, detail
      propensity score, mbsmoke=Nonsmoker
```

Percentiles		Smallest		
1%	.2991634	.2196947		
5%	.544155	.2258079		
10%	.5973879	.2258079	Obs	864
25%	.63777	.2409025	Sum of wgt.	864
50%	.7601717		Mean	.7456264
		Largest	Std. dev.	.1276102
75%	.8453946	.9533503		
90%	.8943686	.9596144	Variance	.0162844
95%	.9096801	.961022	Skewness	-.7701643
99%	.9367017	.9665684	Kurtosis	3.858214

The smallest values do not imply very large weights.

Below we compute and summarize the estimated probabilities of not getting the treatment.

```
. generate pr0 = 1 -pr1
. summarize pr0 if mbsmoke==0, detail
```

pr0				
	Percentiles	Smallest		
1%	.0351884	.0074551		
5%	.0578012	.0079309		
10%	.0674359	.0106305	Obs	3,778
25%	.0950869	.0106305	Sum of wgt.	3,778
50%	.1372589		Mean	.1698913
		Largest	Std. dev.	.1059434
75%	.2211142	.7547572		
90%	.3242757	.774192	Variance	.011224
95%	.3883457	.7803053	Skewness	1.514456
99%	.501537	.7816764	Kurtosis	6.151114

Although there are two small probabilities, overall the small values do not imply large weights.

## Also see

[TE] [teoverlap](#) — Overlap plots

[TE] [teffects aipw](#) — Augmented inverse-probability weighting

[TE] [teffects ipw](#) — Inverse-probability weighting

[TE] [teffects ipwra](#) — Inverse-probability-weighted regression adjustment

[TE] [teffects nnmatch](#) — Nearest-neighbor matching

[TE] [teffects psmatch](#) — Propensity-score matching

[TE] [teffects ra](#) — Regression adjustment

[U] [20 Estimation and postestimation commands](#)

# Title

**teffects psmatch** — Propensity-score matching

<a href="#">Description</a>	<a href="#">Quick start</a>	<a href="#">Menu</a>	<a href="#">Syntax</a>
<a href="#">Options</a>	<a href="#">Remarks and examples</a>	<a href="#">Stored results</a>	<a href="#">Methods and formulas</a>
<a href="#">References</a>	<a href="#">Also see</a>		

## Description

`teffects psmatch` estimates the average treatment effect (ATE) and average treatment effect on the treated (ATET) from observational data by propensity-score matching (PSM). PSM estimators impute the missing potential outcome for each subject by using an average of the outcomes of similar subjects that receive the other treatment level. Similarity between subjects is based on estimated treatment probabilities, known as propensity scores. The treatment effect is computed by taking the average of the difference between the observed and potential outcomes for each subject. `teffects psmatch` accepts a continuous, binary, count, fractional, or nonnegative outcome.

See [\[TE\] teffects intro](#) or [\[TE\] teffects intro advanced](#) for more information about estimating treatment effects from observational data.

## Quick start

ATE of `treat` on `y` estimated by PSM using a logistic model for `treat` on `x` and [indicators](#) for levels of categorical variable `a`

```
teffects psmatch (y) (treat x i.a)
```

As above, but estimate the ATET

```
teffects psmatch (y) (treat x i.a), atet
```

ATE of `treat` using a heteroskedastic probit model for treatment

```
teffects psmatch (y) (treat x i.a, hetprobit(x i.a))
```

With 4 matches per observation

```
teffects psmatch (y) (treat x i.a), nneighbor(4)
```

## Menu

Statistics > Treatment effects > Continuous outcomes > Propensity-score matching

Statistics > Treatment effects > Binary outcomes > Propensity-score matching

Statistics > Treatment effects > Count outcomes > Propensity-score matching

Statistics > Treatment effects > Fractional outcomes > Propensity-score matching

Statistics > Treatment effects > Nonnegative outcomes > Propensity-score matching

## Syntax

```
teffects psmatch (ovar) (tvar tmvarlist [, tmodel]) [if] [in] [weight]
    [, stat options]
```

*ovar* is a binary, count, continuous, fractional, or nonnegative outcome of interest.

*tvar* must contain integer values representing the treatment levels.

*tmvarlist* specifies the variables that predict treatment assignment in the treatment model. Only two treatment levels are allowed.

<i>tmodel</i>	Description
---------------	-------------

---

Model

<code>logit</code>	logistic treatment model; the default
<code>probit</code>	probit treatment model
<code>hetprobit(<i>varlist</i>)</code>	heteroskedastic probit treatment model

---

*tmodel* specifies the model for the treatment variable.

<i>stat</i>	Description
-------------	-------------

---

Stat

<code>ate</code>	estimate average treatment effect in population; the default
<code>atet</code>	estimate average treatment effect on the treated

---

<i>options</i>	Description
Model	
<code>nneighbor(#)</code>	specify number of matches per observation; default is <code>nneighbor(1)</code>
SE/Robust	
<code>vce(vctype)</code>	<i>vctype</i> may be <code>vce(robust [ , nn(#) ])</code> ; use robust Abadie–Imbens standard errors with # matches <code>vce(iid)</code> ; use independent and identically distributed Abadie–Imbens standard errors
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Advanced	
<code>caliper(#)</code>	specify the maximum distance for which two observations are potential neighbors
<code>pstolerance(#)</code>	set tolerance for overlap assumption
<code>osample(newvar)</code>	<i>newvar</i> identifies observations that violate the overlap assumption
<code>control(#   label)</code>	specify the level of <i>tvar</i> that is the control
<code>tlevel(#   label)</code>	specify the level of <i>tvar</i> that is the treatment
<code>generate(stub)</code>	generate variables containing the observation numbers of the nearest neighbors
<code>coeflegend</code>	display legend instead of statistics

*tmvarlist* may contain factor variables; see [U] 11.4.3 **Factor variables**.

*by*, *collect*, and *statsby* are allowed; see [U] 11.1.10 **Prefix commands**.

*fweights* are allowed; see [U] 11.1.6 **weight**.

*coeflegend* does not appear in the dialog box.

See [U] 20 **Estimation and postestimation commands** for more capabilities of estimation commands.

## Options

### Model

`nneighbor(#)` specifies the number of matches per observation. The default is `nneighbor(1)`. Each individual is matched with at least the specified number of individuals from the other treatment level. `nneighbor()` must specify an integer greater than or equal to 1 but no larger than the number of observations in the smallest group.

### Stat

*stat* is one of two statistics: *ate* or *atet*. *ate* is the default.

*ate* specifies that the average treatment effect be estimated.

*atet* specifies that the average treatment effect on the treated be estimated.

## SE/Robust

`vce(vctype)` specifies the standard errors that are reported. By default, `teffects psmatch` uses two matches in estimating the robust standard errors.

`vce(robust [ , nn(#)])` specifies that robust standard errors be reported and that the requested number of matches be used optionally.

`vce(iid)` specifies that standard errors for independent and identically distributed data be reported.

The standard derivative-based standard-error estimators cannot be used by `teffects psmatch`, because these matching estimators are not differentiable. The implemented methods were derived by [Abadie and Imbens \(2006, 2011, 2012\)](#); see [Methods and formulas](#).

As discussed in [Abadie and Imbens \(2008\)](#), bootstrap estimators do not provide reliable standard errors for the estimator implemented by `teffects psmatch`.

## Reporting

`level(#)`; see [\[R\] Estimation options](#).

`display_options`: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [\[R\] Estimation options](#).

## Advanced

`caliper(#)` specifies the maximum distance at which two observations are a potential match. By default, all observations are potential matches regardless of how dissimilar they are.

In `teffects psmatch`, the distance is measured by the estimated propensity score. If an observation has no matches, `teffects psmatch` exits with an error.

`pstolerance(#)` specifies the tolerance used to check the overlap assumption. The default value is `pstolerance(1e-5)`. `teffects` will exit with an error if an observation has an estimated propensity score smaller than that specified by `pstolerance()`.

`osample(newvar)` specifies that indicator variable `newvar` be created to identify observations that violate the overlap assumption. Two checks are made to verify the assumption. The first ensures that the propensity scores are greater than `pstolerance(#)` and less than  $1 - \text{pstolerance}(\#)$ . The second ensures that each observation has at least `nneighbor(#)` matches in the opposite treatment group within the distance specified by `caliper(#)`.

The `vce(robust, nn(#))` option also requires at least `#` matches in the same treatment group within the distance specified by `caliper(#)`.

The average treatment effect on the treated, option `atet`, using `vce(iid)` requires only `nneighbor(#)` control group matches for the treated group.

`control(# | label)` specifies the level of `tvar` that is the control. The default is the first treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(# | label)` specifies the level of `tvar` that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

`generate(stub)` specifies that the observation numbers of the nearest neighbors be stored in the new variables `stub1`, `stub2`, `...`. This option is required if you wish to perform postestimation based

on the matching results. The number of variables generated may be more than `nneighbor(#)` because of tied distances. These variables may not already exist.

The following option is available with `teffects psmatch` but is not shown in the dialog box: `coeflegend`; see [R] [Estimation options](#).

## Remarks and examples

Propensity-score matching uses an average of the outcomes of similar subjects who get the other treatment level to impute the missing potential outcome for each subject. The ATE is computed by taking the average of the difference between the observed and potential outcomes for each subject. `teffects psmatch` determines how near subjects are to each other by using estimated treatment probabilities, known as propensity scores. This type of matching is known as propensity-score matching (PSM).

PSM does not need bias correction, because PSM matches on a single continuous covariate. In contrast, the nearest-neighbor matching estimator implemented in `teffects nnmatch` uses a bias-correction term when matching on more than one continuous covariate. In effect, the PSM estimator parameterizes the bias-correction term in the treatment probability model. See [TE] [teffects intro](#) or [TE] [teffects intro advanced](#) for more information about this estimator.

We will illustrate the use of `teffects psmatch` by using data from a study of the effect of a mother's smoking status during pregnancy (`mbsmoke`) on infant birthweight (`bweight`) as reported by [Cattaneo \(2010\)](#). This dataset also contains information about each mother's age (`age`), education level (`medu`), marital status (`mmarried`), whether the first prenatal exam occurred in the first trimester (`prenatal1`), whether this baby was the mother's first birth (`fbaby`), and the father's age (`fage`).

### ► Example 1: Estimating the ATE

We begin by using `teffects psmatch` to estimate the ATE of `mbsmoke` on `bweight`. We use a logistic model (the default) to predict each subject's propensity score, using covariates `age`, `medu`, `mmarried`, and `fbaby`. Because the performance of PSM hinges upon how well we can predict the propensity scores, we will use factor-variable notation to include both linear and quadratic terms for `age`, the only continuous variable in our model:

```
. use https://www.stata-press.com/data/r17/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects psmatch (bweight) (mbsmoke mmarried c.age##c.age fbaby medu)

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : propensity-score matching      Matches: requested =      1
Outcome model  : matching                      min =      1
Treatment model: logit                        max =      74
```

bweight		AI robust		z	P> z	[95% conf. interval]
		Coefficient	std. err.			
ATE						
	mbsmoke (Smoker vs Nonsmoker)	-210.9683	32.021	-6.59	0.000	-273.7284 -148.2083

The average birthweight if all mothers were to smoke would be 211 grams less than the average that would occur if none of the mothers had smoked.



By default, `teffects psmatch` estimates the ATE by matching each subject to a single subject with the opposite treatment whose propensity score is closest. Sometimes, however, we may want to ensure that matching occurs only when the propensity scores of a subject and a match differ by less than a specified amount. To do that, we use the `caliper()` option. If a match within the distance specified in the `caliper()` option cannot be found, `teffects psmatch` exits.

### ▷ Example 2: Specifying the caliper

Here we reconsider the previous example, first specifying that we only want to consider a pair of observations a match if the absolute difference in the propensity scores is less than 0.03:

```
. teffects psmatch (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu),
> caliper(0.03)
no propensity-score matches for observation 4504 within caliper 0.03; use option
osample() to identify all observations with deficient matches
r(459);
```

The error arose because there is not a smoking mother whose propensity score is within 0.03 of the propensity score of the nonsmoking mother in observation 4504. If we instead raise the caliper to 0.10, we have matches for all subjects and therefore obtain the same results as in [example 1](#):

```
. teffects psmatch (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu),
> caliper(0.1)

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : propensity-score matching      Matches: requested =      1
Outcome model  : matching                        min =      1
Treatment model: logit                          max =      74
```

bweight		AI robust		z	P> z	[95% conf. interval]
Coefficient	std. err.					
ATE						
mbsmoke (Smoker vs Nonsmoker)	-210.9683	32.021	-6.59	0.000	-273.7284	-148.2083

◀

### □ Technical note

[Example 2](#) highlights that estimating the ATE requires finding matches for both the treated and control subjects. In contrast, estimating the ATET only requires finding matches for the treated subjects. Because subject 4504 is a control subject, we can estimate the ATET using `caliper(0.03)`. We must also specify `vce(iid)` because the default robust standard errors for the estimated ATET require viable matches for both treated subjects and control subjects. (This requirement comes from the nonparametric method derived by [Abadie and Imbens \[2012\]](#).)

```
. teffects psmatch (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu),
> atet vce(iid) caliper(0.03)

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : propensity-score matching      Matches: requested =      1
Outcome model  : matching                      min =      1
Treatment model: logit                        max =      74
```

bweight	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
ATET						
mbsmoke (Smoker vs Nonsmoker)	-236.7848	26.11698	-9.07	0.000	-287.9731	-185.5964

□

In the previous examples, each subject was matched to at least one other subject, which is the default behavior for `teffects psmatch`. However, we can request that `teffects psmatch` match each subject to multiple subjects with the opposite treatment level by specifying the `nneighbor()` option. Matching on more distant neighbors can reduce the variance of the estimator at a cost of an increase in bias.

### ▷ Example 3

Now we request that `teffects psmatch` match a mother to four mothers in the opposite treatment group:

```
. teffects psmatch (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu),
> nneighbor(4)

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : propensity-score matching      Matches: requested =      4
Outcome model  : matching                      min =      4
Treatment model: logit                        max =      74
```

bweight	Coefficient	AI robust std. err.	z	P> z	[95% conf. interval]	
ATE						
mbsmoke (Smoker vs Nonsmoker)	-224.006	29.88627	-7.50	0.000	-282.582	-165.43

These results are similar to those reported in [example 1](#).

◁

## Video example

[Treatment effects in Stata: Propensity-score matching](#)

## Stored results

`teffects psmatch` stores the following in `e()`:

### Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level $j$
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(caliper)</code>	maximum distance between matches
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(k_nneighbor)</code>	requested number of matches
<code>e(k_nnmin)</code>	minimum number of matches
<code>e(k_nnmax)</code>	maximum number of matches
<code>e(k_robust)</code>	matches for robust VCE

### Macros

<code>e(cmd)</code>	<code>teffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(subcmd)</code>	<code>psmatch</code>
<code>e(tmodel)</code>	logit, probit, or hetprobit
<code>e(stat)</code>	statistic estimated, <code>ate</code> or <code>atet</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(tlevels)</code>	levels of treatment variable
<code>e(psvvarlist)</code>	variables in propensity-score model
<code>e(hvarlist)</code>	variables for variance, only if <code>hetprobit</code>
<code>e(vce)</code>	<code>vcetype</code> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(datasignature)</code>	the checksum
<code>e(datasignaturevars)</code>	variables used in calculation of checksum
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

### Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance-covariance matrix of the estimators
<code>e(bps)</code>	coefficient vector from propensity-score model
<code>e(Vps)</code>	variance-covariance matrix of the estimators from propensity-score model

### Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

### Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, $p$ -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r`-class command is run after the estimation command.

## Methods and formulas

The methods and formulas used by `teffects psmatch` are documented in the [Methods and formulas](#) of [TE] `teffects nnmatch`.

## References

- Abadie, A., and G. W. Imbens. 2006. Large sample properties of matching estimators for average treatment effects. *Econometrica* 74: 235–267. <https://doi.org/10.1111/j.1468-0262.2006.00655.x>.
- . 2008. On the failure of the bootstrap for matching estimators. *Econometrica* 76: 1537–1557. <https://doi.org/10.3982/ECTA6474>.
- . 2011. Bias-corrected matching estimators for average treatment effects. *Journal of Business and Economic Statistics* 29: 1–11. <https://doi.org/10.1198/jbes.2009.07333>.
- . 2012. Matching on the estimated propensity score. Harvard University and National Bureau of Economic Research. <http://www.nber.org/papers/w15301>.
- Alejo, J., A. F. Galvao, and G. Montes-Rojas. 2020. A practical generalized propensity-score estimator for quantile continuous treatment effects. *Stata Journal* 20: 276–296.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Díaz, J. D., I. Gutiérrez, and J. Rivera. 2021. Implementing `blopmatching` in Stata. *Stata Journal* 21: 180–194.
- Huber, C. 2015. Introduction to treatment effects in Stata: Part 2. *The Stata Blog: Not Elsewhere Classified*. <http://blog.stata.com/2015/08/24/introduction-to-treatment-effects-in-stata-part-2/>.

## Also see

- [TE] **teffects postestimation** — Postestimation tools for teffects
- [TE] **teffects** — Treatment-effects estimation for observational data
- [TE] **teffects nnmatch** — Nearest-neighbor matching
- [U] **20 Estimation and postestimation commands**

# Title

**teffects ra** — Regression adjustment

<a href="#">Description</a>	<a href="#">Quick start</a>	<a href="#">Menu</a>	<a href="#">Syntax</a>
<a href="#">Options</a>	<a href="#">Remarks and examples</a>	<a href="#">Stored results</a>	<a href="#">Methods and formulas</a>
<a href="#">References</a>	<a href="#">Also see</a>		

## Description

`teffects ra` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational data by regression adjustment (RA). RA estimators use contrasts of averages of treatment-specific predicted outcomes to estimate treatment effects. `teffects ra` accepts a continuous, binary, count, fractional, or nonnegative outcome and allows a multivalued treatment.

See [\[TE\] teffects intro](#) or [\[TE\] teffects intro advanced](#) for more information about estimating treatment effects from observational data.

## Quick start

ATE from a linear model of `y1` on `x1` and `x2` with binary treatment `treat2`  
`teffects ra (y1 x1 x2) (treat2)`

As above, but estimate the ATET  
`teffects ra (y1 x1 x2) (treat2), atet`

As above, but estimate the potential-outcome means  
`teffects ra (y1 x1 x2) (treat2), pomeans`

ATE of `treat2` using a heteroskedastic probit model for binary outcome `y2`  
`teffects ra (y2 x1 x2, hetprobit(x1 x2)) (treat2)`

ATE of `treat2` using a Poisson model for count outcome `y3`  
`teffects ra (y3 x1 x2, poisson) (treat2)`

ATE for each level of three-valued treatment `treat3`  
`teffects ra (y1 x1 x2) (treat3)`

As above, and specify that `treat3 = 3` is the control level  
`teffects ra (y1 x1 x2) (treat3), control(3)`

Same as above, specified using the label “MyControl” corresponding to `treat3 = 3`  
`teffects ra (y1 x1 x2) (treat3), control("MyControl")`

## Menu

Statistics > Treatment effects > Continuous outcomes > Regression adjustment

Statistics > Treatment effects > Binary outcomes > Regression adjustment

Statistics > Treatment effects > Count outcomes > Regression adjustment

Statistics > Treatment effects > Fractional outcomes > Regression adjustment

Statistics > Treatment effects > Nonnegative outcomes > Regression adjustment

## Syntax

```
teffects ra (ovar omvarlist [, omodel noconstant]) (tvar) [if] [in] [weight]
           [, stat options]
```

*ovar* is a binary, count, continuous, fractional, or nonnegative outcome of interest.

*omvarlist* specifies the covariates in the outcome model.

*tvar* must contain integer values representing the treatment levels.

<i>omodel</i>	Description
---------------	-------------

### Model

<b>linear</b>	linear outcome model; the default
<b>logit</b>	logistic outcome model
<b>probit</b>	probit outcome model
<b>hetprobit</b> ( <i>varlist</i> )	heteroskedastic probit outcome model
<b>poisson</b>	exponential outcome model
<b>flogit</b>	fractional logistic outcome model
<b>fprobit</b>	fractional probit outcome model
<b>fheterprobit</b> ( <i>varlist</i> )	fractional heteroskedastic probit outcome model

*omodel* specifies the model for the outcome variable.

<i>stat</i>	Description
-------------	-------------

### Stat

<b>ate</b>	estimate average treatment effect in population; the default
<b>atet</b>	estimate average treatment effect on the treated
<b>pomeans</b>	estimate potential-outcome means

<i>options</i>	Description
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>robust</code> , <code>cluster <i>clustvar</i></code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>aequations</code>	display auxiliary-equation results
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
Advanced	
<code>control(#   <i>label</i>)</code>	specify the level of <i>tvar</i> that is the control
<code>tlevel(#   <i>label</i>)</code>	specify the level of <i>tvar</i> that is the treatment
<code>coeflegend</code>	display legend instead of statistics

*omvarlist* may contain factor variables; see [U] 11.4.3 Factor variables.

`bootstrap`, `by`, `collect`, `jackknife`, and `statsby` are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the `bootstrap` prefix; see [R] `bootstrap`.

`fweights`, `iweights`, and `pweights` are allowed; see [U] 11.1.6 `weight`.

`coeflegend` does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

## Options

### Model

`noconstant`; see [R] Estimation options.

### Stat

*stat* is one of three statistics: `ate`, `atet`, or `pomeans`. `ate` is the default.

`ate` specifies that the average treatment effect be estimated.

`atet` specifies that the average treatment effect on the treated be estimated.

`pomeans` specifies that the potential-outcome means for each treatment level be estimated.

### SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster clustvar`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] `vce_option`.

### Reporting

`level(#)`; see [R] Estimation options.

`aequations` specifies that the results for the outcome-model or the treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

*display\_options*: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

---

#### Maximization

*maximize\_options*: `iterate(#)`, `[no]log`, and `from(init_specs)`; see [R] [Maximize](#). These options are seldom used.

*init\_specs* is one of

`matname [ , skip copy ]`

`# [ , # ... ] , copy`

---

#### Advanced

`control(#|label)` specifies the level of *tvar* that is the control. The default is the first treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with statistic `pomeans`. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(#|label)` specifies the level of *tvar* that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

The following option is available with `teffects ra` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

## Remarks and examples

Remarks are presented under the following headings:

[Overview](#)

[Video example](#)

### Overview

Regression adjustment (RA) estimators use the contrasts of the averages of treatment-specific predicted outcomes to estimate treatment effects. RA estimators use a two-step approach to estimating treatment effects:

1. They fit separate regression models of the outcome on a set of covariates for each treatment level.
2. They compute the averages of the predicted outcomes for each subject and treatment level. These averages reflect the POMs. The contrasts of these averages provide estimates of the ATEs. By restricting the computations of the means to the subset of treated subjects, we obtain the ATETs.

RA estimators are consistent as long as the treatment is independent of the potential outcomes after conditioning on the covariates. In fact, `teffects ra` uses an estimation technique that implements both steps at once so that we do not need to correct the standard errors in the second step to reflect the uncertainty surrounding the predicted outcomes.



We will illustrate the use of `teffects ra` by using data from a study of the effect of a mother's smoking status during pregnancy (`mbsmoke`) on infant birthweight (`bweight`) as reported by Cattaneo (2010). This dataset also contains information about each mother's age (`mage`), education level (`medu`), marital status (`mmarried`), whether the first prenatal exam occurred in the first trimester (`prenatal1`), and whether this baby was the mother's first birth (`fbaby`).

### ▷ Example 1: Estimating the ATE

We begin by using `teffects ra` to estimate the average treatment effect of smoking, controlling for first-trimester exam status, marital status, mother's age, and first-birth status. In Stata, we type

```
. use https://www.stata-press.com/data/r17/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ra (bweight prenatal1 mmarried mage fbaby) (mbsmoke)
Iteration 0:   EE criterion = 7.734e-24
Iteration 1:   EE criterion = 1.196e-25
Treatment-effects estimation          Number of obs   =       4,642
Estimator      : regression adjustment
Outcome model  : linear
Treatment model: none
```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
mbsmoke (Smoker vs Nonsmoker)	-239.6392	23.82402	-10.06	0.000	-286.3334	-192.945
POmean						
mbsmoke Nonsmoker	3403.242	9.525207	357.29	0.000	3384.573	3421.911

The average birthweight if all mothers were to smoke would be 240 grams less than the average of 3,403 grams that would occur if none of the mothers had smoked.

◀

The previous results showed us the average amount by which infants' weights are affected by their mothers' decision to smoke. We may instead be interested in knowing the average amount by which the weight of babies born to smoking mothers was decreased as a result of smoking. The `ATET` provides us with the answer.

## ▷ Example 2: Estimating the ATET

To obtain the ATET rather than the ATE, we use the `atet` option:

```
. teffects ra (bweight prenatal1 mmarried mage fbaby) (mbsmoke), atet
Iteration 0: EE criterion = 7.629e-24
Iteration 1: EE criterion = 2.697e-26
Treatment-effects estimation          Number of obs    =    4,642
Estimator      : regression adjustment
Outcome model  : linear
Treatment model: none
```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATET						
mbsmoke (Smoker vs Nonsmoker)	-223.3017	22.7422	-9.82	0.000	-267.8755	-178.7278
POmean						
mbsmoke Nonsmoker	3360.961	12.75749	263.45	0.000	3335.957	3385.966

The average birthweight is 223 grams less when all the mothers who smoke do so than the average of 3,361 grams that would have occurred if none of these mothers had smoked.

The ATET differs from the ATE because the distribution of the covariates among mothers who smoke differs from the distribution for nonsmoking mothers. For example, in [TE] [teffects intro](#), we remarked that in our sample, mothers who smoked tended to be older than those who did not. The differing distributions of covariates also affect the estimated POMs.

◀

By default, `teffects ra` reports the ATE, which is the difference between the two POMs in the case of a binary treatment variable. Sometimes, we want to know the estimated POMs themselves. We might also want to see the actual regression equations used to estimate the POMs. Obtaining this information is easy, as the next example illustrates.

### ▷ Example 3: Estimating the POMs

Here we use the `pomeans` option to display the POMs and the `aequations` option to display the estimated regression coefficients for the treated and untreated subjects.

```
. teffects ra (bweight prenatal1 mmarried mage fbaby) (mbsmoke),
> pomeans aequations
Iteration 0:  EE criterion = 7.734e-24
Iteration 1:  EE criterion = 2.850e-26
Treatment-effects estimation          Number of obs    =    4,642
Estimator      : regression adjustment
Outcome model  : linear
Treatment model: none
```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
<b>POMeans</b>						
mbsmoke						
Nonsmoker	3403.242	9.525207	357.29	0.000	3384.573	3421.911
Smoker	3163.603	21.86351	144.70	0.000	3120.751	3206.455
<b>OME0</b>						
prenatal1	64.40859	27.52699	2.34	0.019	10.45669	118.3605
mmarried	160.9513	26.6162	6.05	0.000	108.7845	213.1181
mage	2.546828	2.084324	1.22	0.222	-1.538373	6.632028
fbaby	-71.3286	19.64701	-3.63	0.000	-109.836	-32.82117
_cons	3202.746	54.01082	59.30	0.000	3096.886	3308.605
<b>OME1</b>						
prenatal1	25.11133	40.37541	0.62	0.534	-54.02302	104.2457
mmarried	133.6617	40.86443	3.27	0.001	53.5689	213.7545
mage	-7.370881	4.21817	-1.75	0.081	-15.63834	.8965804
fbaby	41.43991	39.70712	1.04	0.297	-36.38461	119.2644
_cons	3227.169	104.4059	30.91	0.000	3022.537	3431.801

The nonsmoker POM for infant birthweight is 3,403 grams; that means that if none of the women in our sample smoked during pregnancy, the expected average birthweight would be 3,403 grams. The POM if all mothers did smoke during pregnancy is 3,164 grams, a difference of 240 grams, as we established in [example 1](#). The coefficients for the equation labeled OME0 represent the linear equation used to estimate the nontreated POM, and the coefficients for the equation labeled OME1 represent the linear equation used to estimate the treated POM. The coefficients are identical to those we would obtain using `regress`, but the standard errors differ slightly because `teffects ra` does not make the small-sample adjustment that `regress` does.

◀

We often express statistics as percentages to alleviate scaling issues and aid interpretation. In the present context, we may wish to express an ATE as a percentage of the untreated POM to gain a more intuitive measure of efficacy.

## ▷ Example 4: Reporting the ATE as a percentage

Sometimes, we are interested in reporting the estimated treatment effect as a percentage of the untreated POM. We continue to use the same model as in the previous examples, but we specify the `coeflegend` option so that `teffects ra` reports the names of the parameters. Knowing the correct names to use, we can then use `nlcom` to obtain the percentage change along with its delta-method-based standard error. We type

```
. use https://www.stata-press.com/data/r17/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ra (bweight prenatal1 mmarried mage fbaby) (mbsmoke), coeflegend
Iteration 0:   EE criterion = 7.734e-24
Iteration 1:   EE criterion = 1.196e-25
Treatment-effects estimation           Number of obs   =       4,642
Estimator       : regression adjustment
Outcome model   : linear
Treatment model : none
```

bweight	Coefficient	Legend
ATE		
mbsmoke (Smoker vs Nonsmoker)	-239.6392	_b[ATE:r1vs0.mbsmoke]
POmean		
mbsmoke Nonsmoker	3403.242	_b[POmean:0.mbsmoke]

```
. nlcom _b[ATE:r1vs0.mbsmoke] / _b[POmean:0.mbsmoke]
      _n1_1:  _b[ATE:r1vs0.mbsmoke] / _b[POmean:0.mbsmoke]
```

bweight	Coefficient	Std. err.	z	P> z	[95% conf. interval]
_n1_1	-.070415	.0069245	-10.17	0.000	-.0839867    -.0568433

The average birthweight falls by an estimated 7.0% when every mother smokes relative to the case when no mothers smoke. We also obtain a 95% confidence interval of a 5.7% to 8.4% reduction. ◀

Birthweights cannot be negative, though it is possible for a linear regression model to make negative predictions. A common way to enforce nonnegative predictions is to use an exponential conditional-mean model, which is commonly fitted using the Poisson quasimaximum-likelihood estimator, as discussed in [Cameron and Trivedi \(2005, sec. 5.7\)](#), [Wooldridge \(2010, sec. 18.2\)](#), and [Pawitan \(2001, chap. 14\)](#). `teffects ra` provides an option to use this model rather than linear regression for the outcomes.

### ▷ Example 5: Modeling nonnegative outcomes

Now we refit our model of smoking behavior on birthweight, but we specify the `poisson` option in the outcome-model equation so that `teffects ra` uses the Poisson exponential model rather than linear regression:

```
. teffects ra (bweight prenatal1 mmarried mage fbaby, poisson) (mbsmoke)
Iteration 0:  EE criterion = 3.950e-17
Iteration 1:  EE criterion = 1.244e-23

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : regression adjustment
Outcome model  : Poisson
Treatment model: none
```

		Robust				
	bweight	Coefficient	std. err.	z	P> z	[95% conf. interval]
ATE						
	mbsmoke					
	(Smoker					
	vs					
	Non smoker)	-239.6669	23.83757	-10.05	0.000	-286.3877 -192.9462
POmean						
	mbsmoke					
	Non smoker	3403.178	9.526006	357.25	0.000	3384.508 3421.849

In this case, using a model that forces outcomes to be nonnegative did not make any substantive difference. In this dataset, nearly 90% of babies weigh at least 2,700 grams, and even the smallest baby weighs 340 grams. When the dependent variable is so large, the predictions from Poisson and linear regression models are remarkably similar.

◀

We now consider models for fractional outcomes. Fractional responses concern outcomes between 0 and 1. These responses may be averaged 0/1 outcomes such as participation rates, or they may be variables that are naturally on a 0 to 1 scale such as pollution levels, patient oxygen saturation, and Gini coefficients (income inequality measures).

### ▷ Example 6: Modeling fractional outcomes

We will illustrate the use of `teffects ra` with the outcome-model option `fprobit` by using simulated data. The observations are 543 cities at least 200 miles apart. The data contain information about each city's level of industrialization (`industrial`), average annual rainfall in millimeters (`rainfall`), whether or not the city has a metro or train (`train`), and traffic congestion measured by an index (`traffic`).

Our outcome is the level of pollution (`pollution`) measured on a 0 to 1 scale. Values of `pollution` between 0 and 0.3 have no public health implications, but values greater than 0.7 imply that people with breathing or health problems should remain indoors. We study the effect of a tax on gas-guzzler cars (`guzzler`) on air pollution. A tax that is effective in reducing pollution improves public health.

We estimate the ATE of a gas-guzzler tax on pollution, controlling for average yearly rainfall, traffic congestion, the level of industrialization, and whether the city has a train or a metro by using a fractional probit model.

```

. use https://www.stata-press.com/data/r17/pollution
(Simulated Urban Pollution Data)
. teffects ra (pollution rainfall i.traffic industrial i.train, fprobit) (guzzler)
Iteration 0:  EE criterion = 3.023e-16
Iteration 1:  EE criterion = 9.917e-32
Treatment-effects estimation          Number of obs    =      534
Estimator      : regression adjustment
Outcome model  : fractional probit
Treatment model: none

```

pollution	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
guzzler (tax vs no tax)	-.0960214	.0113896	-8.43	0.000	-.1183447	-.0736981
POmean						
guzzler no tax	.3879346	.0101733	38.13	0.000	.3679952	.407874

The POM if no city were to implement a gas-guzzler tax is an air pollution index of 0.39. If all cities implement a gas-guzzler tax, the air pollution index would decrease by 0.096 relative to a scenario where no city implements the tax.

◀

## Video example

[Treatment effects: Regression adjustment](#)

## Stored results

`teffects ra` stores the following in `e()`:

### Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level $j$
<code>e(N_clust)</code>	number of clusters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(converged)</code>	1 if converged, 0 otherwise

### Macros

<code>e(cmd)</code>	<code>teffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(subcmd)</code>	<code>ra</code>
<code>e(omodel)</code>	<code>linear</code> , <code>logit</code> , <code>probit</code> , <code>hetprobit</code> , <code>poisson</code> , <code>flogit</code> , <code>fprobit</code> , or <code>fhetsprobit</code>
<code>e(stat)</code>	statistic estimated, <code>ate</code> , <code>atet</code> , or <code>pomeans</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<i>vctype</i> specified in <code>vce()</code>
<code>e(vctype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

### Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

### Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

### Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, $p$ -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r`-class command is run after the estimation command.

## Methods and formulas

`teffects ra` implements a [smooth treatment-effects estimator](#). All smooth treatment-effects estimators are documented in [\[TE\] teffects aipw](#).

## References

- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Drukker, D. M. 2016. Exact matching on discrete covariates is the same as regression adjustment. *The Stata Blog: Not Elsewhere Classified*. <http://blog.stata.com/2016/08/16/exact-matching-on-discrete-covariates-is-the-same-as-regression-adjustment/>.
- Huber, C. 2015. Introduction to treatment effects in Stata: Part 1. *The Stata Blog: Not Elsewhere Classified*. <http://blog.stata.com/2015/07/07/introduction-to-treatment-effects-in-stata-part-1/>.
- Pawitan, Y. 2001. *In All Likelihood: Statistical Modelling and Inference Using Likelihood*. Oxford: Oxford University Press.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

## Also see

- [TE] **teffects postestimation** — Postestimation tools for teffects
- [TE] **teffects** — Treatment-effects estimation for observational data
- [ERM] **eregress** — Extended linear regression
- [U] **20 Estimation and postestimation commands**



# Title

**telasso** — Treatment-effects estimation using lasso

<a href="#">Description</a>	<a href="#">Quick start</a>	<a href="#">Menu</a>	<a href="#">Syntax</a>
<a href="#">Options</a>	<a href="#">Remarks and examples</a>	<a href="#">Stored results</a>	<a href="#">Methods and formulas</a>
<a href="#">References</a>	<a href="#">Also see</a>		

## Description

`telasso` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational data by augmented inverse-probability weighting (AIPW) while using lasso methods to select from potential control variables to be included in the model.

AIPW estimators combine aspects of regression-adjustment and inverse-probability-weighted methods. AIPW estimators have the double-robust property.

`telasso` accepts a continuous, binary, count, or nonnegative outcome.

See [\[TE\] teffects intro](#) or [\[TE\] teffects intro advanced](#) for more information about estimating treatment effects from observational data. See [\[LASSO\] Lasso inference intro](#) for more information on estimating coefficients and standard errors for a subset of variables while using lasso methods to select from a set of control variables.

## Quick start

ATE of binary treatment `treat` using a linear model for outcome `y1` on `x1–x100` and a logistic model for `treat` on `w1–w100`; use lassos to select variables from `x1–x100` for the outcome model and from `w1–w100` for the treatment model

```
telasso (y1 x1-x100) (treat w1-w100)
```

As above, but estimate the ATET

```
telasso (y1 x1-x100) (treat w1-w100), atet
```

Use a Poisson model for count outcome `y2`

```
telasso (y2 x1-x100, poisson) (treat w1-w100)
```

As above, but use a probit model for `treat`

```
telasso (y2 x1-x100, poisson) (treat w1-w100, probit)
```

Use BIC instead of a plugin iterative formula to select the optimal  $\lambda^*$  in each lasso

```
telasso (y1 x1-x100) (treat w1-w100), selection(bic)
```

Perform cross-fitting with five folds

```
telasso (y1 x1-x100) (treat w1-w100), xfolds(5)
```

As above, but repeat the cross-fitting procedure 15 times and average the results

```
telasso (y1 x1-x100) (treat w1-w100), xfolds(5) resample(15)
```

Use BIC to select covariates in the lasso for `y1` for treatment level 1 only

```
telasso (y1 x1-x100, lasso(1, selection(bic))) (treat w1-w100)
```

Use cross-validation (CV) for the lasso for `treat` only

```
telasso (y1 x1-x100) (treat w1-w100, lasso(selection(cv)))
```

## Menu

Statistics > Treatment effects > Continuous outcomes > AIPW with lasso covariate selection

Statistics > Treatment effects > Binary outcomes > AIPW with lasso covariate selection

Statistics > Treatment effects > Count outcomes > AIPW with lasso covariate selection

Statistics > Treatment effects > Nonnegative outcomes > AIPW with lasso covariate selection

## Syntax

```
telasso (ovar omvarlist [, omodel om_options])
        (tvar tmvarlist [, tmodel tm_options]) [if] [in] [weight] [, stat options]
```

*ovar* is a binary, count, continuous, or nonnegative outcome of interest.

*omvarlist* specifies the covariates in the outcome model.

*tvar* must contain a binary value representing the treatment levels.

*tmvarlist* specifies the covariates in the treatment model.

<i>omodel</i>	Description
---------------	-------------

Model

<code>linear</code>	linear outcome model; the default
<code>logit</code>	logistic outcome model
<code>probit</code>	probit outcome model
<code>poisson</code>	exponential-mean outcome model

<i>om_options</i>	Description
-------------------	-------------

Model

<code>lasso([#,] <i>lasso_options</i>)</code>	specify options for the lassos for <i>ovar</i> at treatment level #; may be repeated
* <code>sqrtlasso</code>	use square-root lassos instead of lassos for <i>ovar</i>
* <code>sqrtlasso([#,] <i>lasso_options</i>)</code>	specify options for the square-root lassos for <i>ovar</i> at treatment level #; may be repeated
<code>noconstant</code>	suppress constant term
<code>ainclude(<i>varlist</i>)</code>	specify variables that should always be included in the outcome model
<code>noselection</code>	suppress model selection for the outcome model

\* `sqrtlasso()` may be specified only when *omodel* is `linear`.

<i>tmodel</i>	Description
Model	
<code>logit</code>	logistic treatment model; the default
<code>probit</code>	probit treatment model
<i>tm_options</i>	Description
Model	
<code>lasso(<i>lasso_options</i>)</code>	specify options for the lasso for <i>tvar</i>
<code>noconstant</code>	suppress constant term
<code>ainclude(<i>varlist</i>)</code>	specify variables that should always be included in the treatment model
<code>noselection</code>	suppress model selection for the treatment model
<i>stat</i>	Description
Stat	
<code>ate</code>	estimate average treatment effect in population; the default
<code>atet</code>	estimate average treatment effect on the treated
<code>pomeans</code>	estimate potential-outcome means
<i>options</i>	Description
Selection	
<code>selection(plugin)</code>	use a plugin iterative formula to select an optimal value of the lasso penalty parameter $\lambda^*$ for each lasso; the default
<code>selection(cv)</code>	use CV to select an optimal value of the lasso penalty parameter $\lambda^*$ for each lasso
<code>selection(adaptive)</code>	use adaptive lasso to select an optimal value of the lasso penalty parameter $\lambda^*$ for each lasso
<code>selection(bic)</code>	use BIC to select an optimal value of the lasso penalty parameter $\lambda^*$ for each lasso
<code>xfolds[#]</code>	use # folds for cross-fitting
<code>resample[#]</code>	repeat sample splitting # times and average results
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>robust</code> or <code>cluster <i>clustvar</i></code>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Optimization	
<code>rseed(#)</code>	set random-number seed
<code>verbose</code>	display a verbose iteration log
<code>[no]log</code>	display or suppress an iteration log

Advanced

<code>pstolerance(#)</code>	set tolerance for overlap assumption
<code>osample(newvar)</code>	<i>newvar</i> identifies observations that violate the overlap assumption
<code>control(#   label)</code>	specify the level of <i>tvar</i> that is the control
<code>reestimate</code>	refit the model after using <code>lassoselect</code> to select a different $\lambda^*$
<code>noheader</code>	do not display the header on the coefficient table
<code>coeflegend</code>	display legend instead of statistics

*omvarlist* and *tmvarlist* may contain factor variables; see [U] 11.4.3 Factor variables.

`collect` is allowed; see [U] 11.1.10 Prefix commands.

Default weights are not allowed. `iweights` are allowed when `selection(plugin)`, `selection(adaptive)`, `selection(cv)`, or `selection(bic)` is specified. `fweights` are allowed when `selection(plugin)` or `selection(bic)` is specified. See [U] 11.1.6 weight.

`reestimate`, `noheader`, and `coeflegend` do not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

## Options

### Model

Options specific to the outcome model (*om\_options*) are the following:

`lasso([#, ] lasso_options)` lets you set different options for different lassos or set advanced options for all lassos for the outcome model. Specify a `#` followed by the options to apply settings to the lasso of *ovar* at treatment level `#` only. This option is repeatable as long as a different treatment level is given in each specification. If `#` is not specified, the lasso options apply to lassos of *ovar* at all treatment levels. *lasso\_options* are `selection()`, `grid()`, `stop()`, `cvtolerance()`, `bictolerance()`, `tolerance()`, and `dtolerance()`; see [LASSO] lasso options. When `lasso([#, ] selection())` is specified, it overrides any global `selection()` option for the outcome variable at the specified treatment level.

`sqrtilasso` specifies that square-root lassos rather than lassos should be used in selecting variables in models for the outcome at both treatment levels. This option is not allowed if *omodel* is `logit`, `probit`, or `poisson`.

`sqrtilasso([#, ] lasso_options)` works like the option `lasso()`, except square-root lassos for the outcome variable are done rather than regular lassos. This option is not allowed if *omodel* is `logit`, `probit`, or `poisson`. Specify a `#` followed by the options to apply settings to the lasso of *ovar* at treatment level `#` only. This option is repeatable as long as a different treatment level is given in each specification. If `#` is not specified, the lasso options apply to lassos of *ovar* at all treatment levels. *lasso\_options* are `selection()`, `grid()`, `stop()`, `cvtolerance()`, `bictolerance()`, `tolerance()`, and `dtolerance()`; see [LASSO] lasso options. When `sqrtilasso([#, ] selection())` is specified, it overrides any global `selection()` option for the outcome variable at the specified treatment level.

`noconstant` suppresses the constant term in the outcome model.

`ainclude(varlist)` specifies variables that are always included in the outcome model. Lassos will choose to include or exclude other variables in *omvarlist*.

`noselection` specifies not to perform model selection in the outcome model.

Options specific to the treatment model (*tm\_options*) are the following:

`lasso(lasso_options)` lets you set advanced options for the lasso for the treatment model. *lasso\_options* are `selection()`, `grid()`, `stop()`, `cvtolerance()`, `bictolerance()`, `tolerance()`, and `dtolerance()`; see [LASSO] [lasso options](#). When `lasso(selection())` is specified, it overrides any global `selection()` option for the treatment variable.

`noconstant` suppresses the constant term in the treatment model.

`ainclude(varlist)` specifies variables that are always included in the treatment model. Lassos will choose to include or exclude other variables in *tmvarlist*.

`noselection` specifies not to perform model selection in the treatment model.

#### Selection

`selection(plugin | cv | adaptive | bic)` specifies the selection method for choosing an optimal value of the lasso penalty parameter  $\lambda^*$  for each lasso or square-root lasso estimation. Separate lassos are estimated for the outcome variable at each treatment level and for the treatment variable. Specifying `selection()` changes the selection method for all these lassos. You can specify different selection methods for different lassos by using `lasso()` or `sqrtlasso()` in the outcome model specification or the option `lasso()` in the treatment model specification. When `lasso()` or `sqrtlasso()` is used to specify a different selection method for the lassos of outcome or treatment variables, they override the global setting made using `selection()` for the specified model.

`selection(plugin)` is the default. It selects  $\lambda^*$  based on a “plugin” iterative formula dependent on the data. See [LASSO] [lasso options](#).

`selection(cv)` selects the  $\lambda^*$  that gives the minimum of the CV function. See [LASSO] [lasso options](#).

`selection(adaptive)` selects  $\lambda^*$  using the adaptive lasso selection method. It cannot be specified when `sqrtlasso()` is specified for the outcome model. See [LASSO] [lasso options](#).

`selection(bic)` selects the  $\lambda^*$  that gives the minimum of the BIC function. See [LASSO] [lasso options](#).

`xfolds[#]` specifies the number of folds for cross-fitting. Not specifying `xfolds` or `xfolds(#)` is equivalent to specifying `xfolds(1)`. In other words, by default no cross-fitting is done. Specifying `xfolds` alone is equivalent to specifying `xfolds(10)`; that is, cross-fitting is done by randomly dividing the original data into 10 folds.

`resample[#]` specifies that sample splitting be repeated and results averaged. This reduces the effects of the randomness of sample splitting on the estimated coefficients. Not specifying `resample` or `resample(#)` is equivalent to specifying `resample(1)`. In other words, by default no resampling is done. Specifying `resample` alone is equivalent to specifying `resample(10)`; that is, sample splitting is repeated 10 times. For each sample split, lassos are computed. So when this option is not specified, lassos are repeated `xfolds(#)` times. But when `resample(#)` is specified, lassos are repeated `xfolds(#)`  $\times$  `resample(#)` times. Thus, while we recommend using `resample` to get final results, note that it can be an extremely time-consuming procedure.

#### SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`) and that allow for intragroup correlation (`cluster clustvar`); see [R] [vce\\_option](#).

## Reporting

`level(#)`; see [R] [Estimation options](#).

`display_options`: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

## Optimization

`rseed(#)` sets the random-number seed. This option can be used to reproduce results. `rseed(#)` is equivalent to typing `set seed #` prior to running `telasso`. Random numbers are used to produce split samples for cross-fitting. So for all `selection()` options, if you want to reproduce your results, you must either use this option or use `set seed`. See [R] [set seed](#).

`verbose` displays a verbose log showing the iterations of each lasso estimation. This option is useful when doing `selection(cv)` or `selection(adaptive)`. It allows monitoring the progress of the lasso estimations for these selection methods, which can be time consuming when there are many variables specified in `omvarlist` or `tmvarlist`.

`[no]` `log` displays or suppresses a log showing the progress of the estimation. By default, one-line messages indicating when each lasso estimation begins are shown. Specify `verbose` to see a more detailed log.

## Advanced

`pstolerance(#)` specifies the tolerance used to check the overlap assumption. The default value is `pstolerance(1e-5)`. `telasso` will exit with an error if an observation has an estimated propensity score smaller than that specified by `pstolerance()`.

`osample(newvar)` specifies that indicator variable `newvar` be created to identify observations that violate the overlap assumption.

`control(#|label)` specifies the level of `tvar` that is the control. The default is the first treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with statistic `pomeans`.

The following options are available with `telasso` but are not shown in the dialog box:

`reestimate` is an advanced option that refits the `telasso` model based on changes made to the underlying lassos using `lassoselect`. After running `telasso`, you can select a different  $\lambda^*$  for one or more of the lassos estimated by `telasso`. After selecting  $\lambda^*$ , you type `telasso`, `reestimate` to refit the `telasso` model based on the newly selected  $\lambda$ 's.

`reestimate` may be combined only with reporting options.

`noheader` prevents the coefficient table header from being displayed.

`coeflegend`; see [R] [Estimation options](#).

## Remarks and examples

Remarks are presented under the following headings:

[Overview](#)

[Estimating the ATE with lassos for covariate selection](#)

[Choosing the tuning parameter](#)

[Estimating the ATET](#)

[High-dimensional semiparametric models](#)

## Overview

`telasso` estimates treatment effects while using lasso techniques to select the variables in the model. Like the `lasso for inference` commands, inferences drawn about the ATEs, ATETs, and POMs reported by `telasso` are robust to model selection mistakes made by lasso. In addition, like the `teffects aipw` command, the estimated effects are consistent if the functional form of only one of the outcome or treatment model is correctly specified. Thus, `telasso` is a version of the AIPW estimator that also allows for covariate selection via lasso and is robust to functional form misspecification in one of the models as well as errors in selection of covariates by lasso. It is, however, required that the true covariates are a subset of the specified list of potential covariates. See [TE] [teffects intro](#) or [TE] [teffects intro advanced](#) for an introduction of treatment effects estimation. See [TE] [teffects aipw](#) for more information on the AIPW estimator. See [LASSO] [Lasso intro](#) for an introduction to lasso. See [LASSO] [Lasso inference intro](#) for an introduction to making inference after using lasso for model selection.

Researchers who have datasets with many variables, sometimes more than observations, face difficult decisions. They cannot use all the variables in their dataset as covariates in a model, and therefore, they may want to use a model selection method such as lasso to select covariates. In addition, rather than simply allowing covariates to enter a model linearly, researchers may want to fit a more flexible and realistic model to their data by including higher-order terms, interactions, spline basis functions, and the like, but this can also produce more terms than can be included in a model and require a method such as lasso for selection. The lasso methods used in `telasso` allow for making valid inference after selecting from among the potential covariates.

In the context of treatment-effect estimation, why do we need model selection? It is because of the intrinsic conflicts between two crucial assumptions used to identify treatment effects: conditional independence (CI) and overlap assumptions. CI means that, dependent on a set of control variables, the potential outcome is independent of the treatment assignment. The more variables there are in the model, the more plausible it is that the CI assumption is satisfied. On the other hand, the overlap assumption implies that there is always a positive probability that any given unit is treated or untreated. The fewer variables there are in the model, the more comfortable we can be that the overlap assumption is satisfied. To summarize, the conflict is that the CI assumption expects many variables in the model, but the overlap assumption expects few variables. By including many potential variables in the model and allowing lasso to select from among them, we can reconcile this conflict. For a more detailed discussion, see [Farrell \(2015\)](#) and [Chernozhukov et al. \(2018\)](#).

Model selection, however, does not come for free. If researchers use model selection but conduct inference ignoring the fact that they did model selection, the inference results are possibly wrong. This is because model selection techniques make mistakes. Making inferences without considering the variability in model selection is a dangerous practice (see [Leeb and Pötscher \[2005\]](#) and [Leeb and Pötscher \[2006\]](#)). Instead, the inference should be robust to model selection mistakes.

Recently, [Farrell \(2015\)](#) and [Chernozhukov et al. \(2018\)](#) proposed that modern model selection techniques, such as lasso methods for inference, be combined with the doubly robust AIPW estimator in [Rosenbaum and Rubin \(1983\)](#) to estimate the ATEs. The intuition behind this method is twofold. On one hand, lasso techniques resolve the conflicts between the CI and overlap assumptions. Although CI assumption expects many variables, we only need the covariates that matter for the outcome. If lasso selects a sparse model, the overlap assumption is more plausible to be satisfied. On the other hand, the proposed estimator should guard against model selection mistakes. The doubly robust AIPW estimator happens to satisfy this property. Double robustness also means that the estimates of treatment effects are consistent if either the outcome model or the treatment model is correctly specified.

`telasso` implements estimators in [Chernozhukov et al. \(2018\)](#) for the ATEs, ATETs, and POMs from observational data.

The `telasso` estimator uses a five-step approach to estimating ATE:

1. It uses lasso techniques to select variables in the outcome model for each treatment level.
2. Based on the selected variables in step 1, it fits separate regression models of the outcome for each treatment level and obtains the treatment-specific predicted outcomes for each subject.
3. It uses lasso techniques to select variables in the treatment model.
4. Based on the selected variables in step 3, it estimates the parameters of the treatment model and computes inverse-probability weights.
5. It computes the weighted means of the treatment-specific predicted outcomes, where the weights are the inverse-probability weights computed in step 4. The contrasts of these weighted averages provide the estimates of the ATE.

Steps 1 and 3 perform the model selection for the outcome and treatment models, respectively. Using the selected variables, steps 2, 4, and 5 construct moment conditions to estimating ATEs. The resulting estimator is consistent under CI, overlap, and independent and identically distributed assumptions. The inference is robust to the mild model selection mistakes that could happen in steps 1 and 3. This estimator is also robust to model misspecification in either the outcome or the treatment model because of the double robust moment condition used in step 5.

`telasso` also implements the double machine learning estimator in [Chernozhukov et al. \(2018\)](#) for ATEs, ATETs, and POMs. Double machine learning relaxes the sparsity assumption needed for lasso methods. The sparsity assumption implies that lasso can only have good properties if it selects a few variables from a potentially large number of candidates. Double machine learning allows `telasso` to select more variables in the models and still be valid.

`telasso` allows the outcome variable to be modeled using a linear, logistic, probit, or Poisson model. For a linear outcome model, square-root lasso can be used instead of lasso in step 1.

## Estimating the ATE with lassos for covariate selection

In some cases, we want to estimate a treatment effect when we have several variables in our dataset that, when interacted, create a large set of candidate covariates in our model.

### ► Example 1: ATE of bilateral lung transplant

We first illustrate `telasso` with an example that compares two types of lung transplants. Bilateral lung transplant (BLT) is usually associated with a higher death rate in the short term after the operation but with a more significant improvement in the quality of life compared with the single lung transplant (SLT). As a result, for patients who need to decide between these two treatment options, knowing the effect of BLT (versus SLT) on quality of life is essential. We can measure quality of life based on an individual's forced expiratory volume in one second (FEV1).

We have a fictional dataset (`lung.dta`) inspired by [Koch, Vock, and Wolfson \(2018\)](#). The outcome (`fev1p`) is FEV1% measured one year after the operation. FEV1% is the percentage of FEV1 that the patient has relative to a healthy person with similar characteristics. The treatment variable (`transtype`) indicates whether the treatment is BLT or SLT.



To open the dataset and describe it, we type

```
. use https://www.stata-press.com/data/r17/lung
(Fictional data on lung transplant)
. describe *, short
```

Variable name	Storage type	Display format	Value label	Variable label
agep	byte	%10.0g		Patient age (years)
bmip	double	%10.0g		Patient body mass index
diabetesp	byte	%12.0g	lbdiab	Patient diabetes status
heightp	double	%10.0g		Patient height (cm)
o2amt	double	%10.0g		Oxygen delivered
karn	byte	%8.0g	lbytes	Karnofsky score > 60
lungals	double	%10.0g		Lung allocation score
racep	byte	%8.0g	lbrace	Patient race
sexp	byte	%8.0g	lbsex	Patient gender
lifesvent	byte	%8.0g	lbytes	Life support ventilator needed
assistvent	byte	%8.0g	lbytes	Assisted ventilation needed
centervol	double	%10.0g		Center volume
walkdist	double	%10.0g		Walking distance in 6 minutes
o2rest	byte	%8.0g	lbytes	Oxygen needed at rest
aged	byte	%10.0g		Donor age (years)
raced	byte	%8.0g	lbrace	Donor race
bmid	double	%10.0g		Donor body mass index
smoked	byte	%8.0g	lbytes	Donor if has history of smoking
cmv	byte	%8.0g	lbytes	Positive cytomegalovirus test
deathcause	byte	%8.0g	lbytes	Cause of death - traumatic brain injury
diabetesd	byte	%12.0g	lbdiab	Donor diabetes status
expandd	byte	%8.0g	lbytes	Expanded donor needed
heightd	double	%10.0g		Donor height (cm)
sexd	byte	%8.0g	lbsex	Donor gender
distd	int	%10.0g		Donor to treatment center distance
lungpo2	double	%10.0g		Lung P02
lungalloc	byte	%8.0g	lballo	Lung allocation status
hratio	double	%10.0g		Height ratio
ischemict	double	%10.0g		Ischemic time
genderm	byte	%19.0g	lbgm	Matching gender status
racem	byte	%17.0g	lbrm	Matching race status
transtype	byte	%8.0g	lbtau	Lung transplant type
fev1p	double	%10.0g		Percentage of predicted value of FEV1

Thirty-one variables measure some characteristics of the patients and donors. To construct control variables, we want to use these variables and the interactions among them. It would be tedious to type these variable names one by one to distinguish between continuous and categorical variables. `v1` is a suite of commands that simplifies this process.

First, we use `v1 set` to automatically partition the variables into continuous and categorical variables. The global macro `$v1categorical` contains all the categorical variable names, and `$v1continuous` contains all the continuous variable names.

```
. vl set
```

Macro	Macro's contents	
	# Vars	Description
System		
\$vlcategorical	18	categorical variables
\$vlcontinuous	13	continuous variables
\$vluncertain	2	perhaps continuous, perhaps categorical variables
\$vlother	0	all missing or constant variables

#### Notes

1. Review contents of `vlcategorical` and `vlcontinuous` to ensure they are correct. Type `vl list vlcategorical` and type `vl list vlcontinuous`.
2. If there are any variables in `vluncertain`, you can reallocate them to `vlcategorical`, `vlcontinuous`, or `vlother`. Type `vl list vluncertain`.
3. Use `vl move` to move variables among classifications. For example, type `vl move (x50 x80) vlcontinuous` to move variables `x50` and `x80` to the continuous classification.
4. `vlnames` are global macros. Type the `vlname` without the leading dollar sign (\$) when using `vl` commands. Example: `vlcategorical` not `$vlcategorical`. Type the dollar sign with other Stata commands to get a `varlist`.

```
. display "$vlcontinuous"
bmip heightp o2amt lungals centervol walkdist bmid heightd distd lungpo2
> hratio ischemict fev1p

. display "$vlcategorical"
diabetesp karn racep sexp lifesvent assisvent o2rest raced smoked cmv
> deathcause diabetesd expandd sexd lungalloc genderm racem transtype
```

Second, we use `vl create` to create customized variable lists. Specifically, `$cvars` contains all the continuous variables except the outcome (`fev1p`), and `$fvars` consists of all the categorical variables except the treatment (`transtype`). Finally, `vl sub` substitutes the global macro `$allvars` with the full second-order interaction between the continuous variables in `$cvars` and categorical variables in `$fvars`. We will use `$allvars` as the control variables for both outcome model and treatment model.

```
. vl create cvars = vlcontinuous - (fev1p)
note: $cvars initialized with 12 variables.

. vl create fvars = vlcategorical - (transtype)
note: $fvars initialized with 17 variables.

. vl sub allvars = c.cvars i.fvars c.cvars#i.fvars
```

Now we are ready to use `telasso` to estimate the ATEs. We assume a linear outcome model and a logit treatment model, the defaults for `telasso`. We type

```
. telasso (fev1p $allvars) (transtype $allvars)
Estimating lasso for outcome fev1p if transtype = 0 using plugin method ...
Estimating lasso for outcome fev1p if transtype = 1 using plugin method ...
Estimating lasso for treatment transtype using plugin method ...
Estimating ATE ...

Treatment-effects lasso estimation      Number of observations      =      937
Outcome model: linear                   Number of controls         =      454
Treatment model: logit                  Number of selected controls =       8
```

fev1p	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE transtype (BLT vs SLT)	37.51841	.1606703	233.51	0.000	37.20351	37.83332
POmean transtype SLT	46.4938	.2021582	229.99	0.000	46.09757	46.89002

The FEV1% if all the patients were to choose BLT is expected to be 38 percentage points higher than the 46% average expected if all patients were to choose a SLT. Among the 454 control variables, `telasso` selects only 8 of them. To summarize the model selection for both the outcome and the treatment models, we can use `lassoinfo`.

```
. lassoinfo
Estimate: active
Command: telasso
```

Variable	Model	Selection method	lambda	No. of selected variables
fev1p	linear	plugin	.2239121	5
transt~e ~0	linear	plugin	.1986153	6
transt~e ~1	logit	plugin	.0748279	3

If we want to see which variables are selected by each lasso, we can use `lassocoeff`. Notice that there are two lassos associated with the outcome of `fev1p`. One is for outcome `fev1p` when the treatment `transtype` is 0, and the other is for `fev1p` when `transtype` is 1. So, in `lassocoeff`, we need to use the options `tlevel()` and `for()` to refer to the lasso for the outcome variable at a specified treatment level. In contrast, for the treatment of `transtype`, there is only one lasso; therefore, we only need to use option `for()` to specify the lasso for the treatment variable.

```
. lassocoef (., for(fev1p) tlevel(0)) (., for(fev1p) tlevel(1))
> (., for(transtype))
```

	fev1p(0)	fev1p(1)	transtype
heightp	x	x	
centervol	x	x	
walkdist	x	x	x
lungpo2	x	x	x
diabetesd#c.lungpo2			
0	x		
diabetesp#c.walkdist			
0		x	
assisvent#c.walkdist			
0		x	
ischemict			x
_cons	x	x	x

Legend:

b - base level  
e - empty cell  
o - omitted  
x - estimated

The lassos selected `walkdist` and `lungpo2` in all three models, while `heightp`, `centervol`, `0.diabetesd#c.lungpo2`, `0.diabetesp#c.walkdist`, `0.assisvent#c.walkdist`, and `ischemict` were each selected as covariates in one or two models.

## Choosing the tuning parameter

By default, `telasso` uses a plugin method to choose the tuning parameter  $\lambda$  in the lasso steps. We can also use BIC, cross-validation, or adaptive lasso to select the optimal  $\lambda$ .

### ► Example 2: Choosing $\lambda$ via BIC

Here we use the `selection(bic)` option to select  $\lambda$  by minimizing BIC.

```
. telasso (fevip $allvars) (transtype $allvars), selection(bic)
Estimating lasso for outcome fevip if transtype = 0 using BIC ...
Estimating lasso for outcome fevip if transtype = 1 using BIC ...
Estimating lasso for treatment transtype using BIC ...
Estimating ATE ...

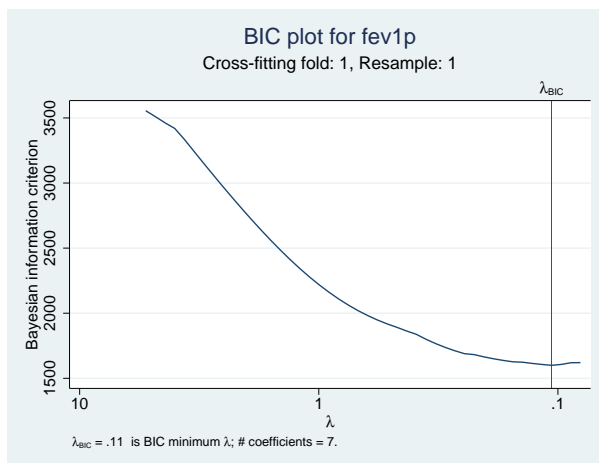
Treatment-effects lasso estimation   Number of observations   =   937
Outcome model: linear                Number of controls      =   454
Treatment model: logit               Number of selected controls =   18
```

fevip	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE transtype (BLT vs SLT)	37.54872	.2222001	168.99	0.000	37.11322	37.98423
POmean transtype SLT	46.44739	.2282797	203.47	0.000	45.99997	46.89481

We can interpret the estimation results in a similar way as in [example 1](#). The FEV1% if all the patients were to choose BLT is expected to be 38 percentage points higher than the average of 46% that would be expected if all patients were to choose a SLT. This result is similar to [example 1](#), where we used the default plugin method to select the tuning parameter. However, among 454 controls, `telasso` with BIC selects 18 of them, which is more than the plugin method selected.

We can use `bicplot` for a global view about how the BIC function changes as the  $\lambda$ 's change. Here we show the BIC plot for the outcome `fev1p` when treatment `transtype` is 1. In `bicplot`, we use options `for()` and `tlevel()` to refer to the lasso for the outcome variable at a specified treatment level.

```
. bicplot, for(fev1p) tlevel(1)
```



From this plot, we can see that the optimal  $\lambda = 0.11$  is chosen by the minimum BIC. If we want to investigate this further, we can use `lassoknots` to see which variables are selected or dropped for each  $\lambda$ .

```
. lassoknots, for(fev1p) tlevel(1) display(nonzero var bic)
```

ID	lambda	No. of nonzero coef.	BIC	Variables (A)dded, (R)emoved, or left (U)nchanged
2	4.823808	1	3510.118	A walkdist
5	3.649034	2	3335.87	A lungpo2
28	.4294227	3	1863.595	A heightp
29	.391274	4	1838.266	A centervol
35	.2239014	6	1682.065	A 0.diabetesp#c.walkdist 0.assisvent#c.walkdist
40	.1406166	7	1623.74	A heightd
* 43	.1063713	7	1599.977	U
44	.0969216	9	1606.419	A 0.karn#c.walkdist 0.raced#c.lungpo2
45	.0883113	12	1619.473	A 0.sex#centervol 1.racep#c.centervol 0.deathcause#c.centervol
46	.080466	13	1620.666	A 1.lungalloc#c.lungpo2

\* lambda selected by Bayesian information criterion.

We see that the 43rd  $\lambda$ , with value 0.1064, minimizes the BIC function, and there are seven selected variables at this  $\lambda$ .

## Estimating the ATET

Sometimes, we want to estimate the ATETs to determine the effect on those who actually received the treatment.

### ► Example 3: ATET for BLT

We use the `atet` option to estimate the ATETs. We type

```
. telasso (fevip $allvars) (transtype $allvars), atet
Estimating lasso for outcome fevip if transtype = 0 using plugin method ...
Estimating lasso for outcome fevip if transtype = 1 using plugin method ...
Estimating lasso for treatment transtype using plugin method ...
Estimating ATET ...

Treatment-effects lasso estimation      Number of observations      =      937
Outcome model: linear                   Number of controls         =      454
Treatment model: logit                  Number of selected controls =        8
```

fevip	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATET transtype (BLT vs SLT)	35.78157	.1831478	195.37	0.000	35.42261	36.14053
P0mean transtype SLT	43.35214	1.268976	34.16	0.000	40.86499	45.83929

For the patients who have a BLT, we expect the average FEV1% to be 36 percentage points higher than if all of them choose an SLT.



## High-dimensional semiparametric models

Sometimes, the theory suggests that some variables are essential to control the confounders, but it is silent on the functional form of the model. In the next example, we will illustrate how to use `telasso` to estimate the ATE in a high-dimensional semiparametric model.

### ► Example 4: ATE of 401(k) eligibility

We want to estimate the effect of 401(k) eligibility (`e401`) on net financial assets (`asset`) using data reported by [Chernozhukov and Hansen \(2004\)](#). These data are from a sample of households in the 1990 Survey of Income and Program Participation (SIPP). The data contain information on the head of the household: income (`income`), age (`age`), years of education (`educ`), whether to receive pension benefit (`pension`), marital status (`married`), and whether to participate in the IRA (`ira`).

One concern when determining the effect of 401(k) eligibility on financial assets is that choosing to work for a company that offers a 401(k) plan is not randomly assigned. To overcome this issue, [Poterba, Venti, and Wise \(1994, 1995\)](#) propose that after conditioning on income, we can take working for a company that offers a 401(k) as exogenous.

We want to include `income`, `age`, and `educ` as covariates in the model, but we do not want to assume that they enter the model linearly. Because we want a more flexible model, we use `mkspline` to create linear splines with knots equally spaced or at percentiles of data for `income`, `age`, and `educ`. The generated variables will be used to form a semiparametric model. These variables have the common stub `_b` for easy reference later.

```
. use https://www.stata-press.com/data/r17/assets, clear
(Excerpt from Chernozhukov and Hansen (2004))
. mkspline _bincome 5 = income
. mkspline _bage    5 = age
. mkspline _beduc   5 = educ
```

We can look at the generated linear spline variables by describing them. We see that for each variable of `income`, `age`, and `educ`, five terms are generated.

```
. describe _b*
```

Variable name	Storage type	Display format	Value label	Variable label
<code>_bincome1</code>	float	%9.0g		income: (.,48424.8)
<code>_bincome2</code>	float	%9.0g		income: (48424.8,96849.6)
<code>_bincome3</code>	float	%9.0g		income: (96849.60000000001,145274.4)
<code>_bincome4</code>	float	%9.0g		income: (145274.4,193699.2)
<code>_bincome5</code>	float	%9.0g		income: (193699.2,.)
<code>_bage1</code>	float	%9.0g		age: (.,32.8)
<code>_bage2</code>	float	%9.0g		age: (32.8,40.6)
<code>_bage3</code>	float	%9.0g		age: (40.599999999999999,48.4)
<code>_bage4</code>	float	%9.0g		age: (48.399999999999998,56.2)
<code>_bage5</code>	float	%9.0g		age: (56.199999999999997,.)
<code>_beduc1</code>	float	%9.0g		educ: (.,4.4)
<code>_beduc2</code>	float	%9.0g		educ: (4.4,7.8)
<code>_beduc3</code>	float	%9.0g		educ: (7.800000000000001,11.2)
<code>_beduc4</code>	float	%9.0g		educ: (11.2,14.6)
<code>_beduc5</code>	float	%9.0g		educ: (14.6,.)

Next, we define the control variables by interacting the categorical controls (`pension`, `married`, and `ira`) and the generated spline variables (`_b*`). The global macro `$controls` contains the defined control variables, and we will use it in both the outcome and the treatment model.

```
. global vars c(_b*) i.(pension married ira)
. global controls $vars ($vars)#($vars)
```



Now, we are ready to use `telasso` to estimate the ATEs of 401(k) eligibility on net financial assets. We use a linear nonparametric series to approximate the outcome model. Moreover, we use a logit nonparametric series to approximate the treatment model. The global macro `$controls` defines terms in the nonparametric series.

```
. telasso (assets $controls) (e401 $controls)
Estimating lasso for outcome assets if e401k = 0 using plugin method ...
Estimating lasso for outcome assets if e401k = 1 using plugin method ...
Estimating lasso for treatment e401k using plugin method ...
Estimating ATE ...

Treatment-effects lasso estimation      Number of observations      =      9,913
Outcome model: linear                   Number of controls          =       243
Treatment model: logit                  Number of selected controls =        26
```

assets	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
e401k (Eligible vs Not elig..)	7850.285	1092.952	7.18	0.000	5708.139	9992.431
POmean						
e401k Not eligi..	13893.49	785.1038	17.70	0.000	12354.72	15432.27

The net financial assets if all the workers work for the companies with a 401(k) plan is expected to be \$7,850 more than the average of \$13,839 that is expected if all the workers work for the companies without a 401(k) plan.

4

## Stored results

`telasso` stores the following in `e()`:

### Scalars

- `e(N)` number of observations
- `e(NO)` number of observations for treatment level 0
- `e(N1)` number of observations for treatment level 1
- `e(N_clust)` number of clusters
- `e(k_omvars)` number of potential control variables in the outcome model
- `e(k_omvars_sel)` number of selected control variables in the outcome model
- `e(k_tmvars)` number of potential control variables in the treatment model
- `e(k_tmvars_sel)` number of selected control variables in the treatment model
- `e(k_controls)` number of potential control variables in the outcome and treatment models
- `e(k_controls_sel)` number of selected control variables in the outcome and treatment models
- `e(k_levels)` number of levels in treatment variable
- `e(n_xfolds)` number of folds for cross-fitting
- `e(n_resample)` number of resamples
- `e(treated)` level of treatment variable defined as treated
- `e(control)` level of treatment variable defined as control
- `e(converged)` 1 if converged, 0 otherwise

### Macros

- `e(cmd)` `telasso`
- `e(cmdline)` command as typed
- `e(depvar)` name of outcome variable

<code>e(tvar)</code>	name of treatment variable
<code>e(tmodel)</code>	logit or probit
<code>e(omodel)</code>	linear, logit, probit, or poisson
<code>e(omvars)</code>	potential control variables in the outcome model
<code>e(omvars_sel)</code>	selected control variables in the outcome model
<code>e(tmvars)</code>	potential control variables in the treatment model
<code>e(tmvars_sel)</code>	selected control variables in the treatment model
<code>e(stat)</code>	statistic estimated: <code>ate</code> , <code>atet</code> , or <code>pomeans</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(rngstate)</code>	random-number state used
<code>e(properties)</code>	<code>b V</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(select_cmd)</code>	program used to implement <code>lassoselect</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>
Matrices	
<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators
Functions	
<code>e(sample)</code>	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r`-class command is run after the estimation command.

## Methods and formulas

Methods and formulas are presented under the following headings:

- The model*
- Neyman orthogonal moments*
- Double machine learning*
- Resampling the partitions*

## The model

We consider estimating the ATEs, ATETs, and POMs when the treatment effects are heterogeneous and the treatment is binary. For notational simplicity, we drop the subscript *i* indicating the *i*th observation.

The outcome model is

$$y = g_0(\tau, \mathbf{x}) + u \quad E(u|\tau, \mathbf{x}) = 0$$

where *y* is the outcome variable,  $\tau$  is the binary treatment variable,  $\mathbf{x}$  are potentially high-dimensional control variables in the outcome model, and  $g_0(\tau, \mathbf{x})$  is the expected potential outcome given a level of treatment and covariates  $\mathbf{x}$ . Because each unit can only be treated or not treated, the observed outcome *y* can only be one of  $g_0(1, \mathbf{x}) + u$  or  $g_0(0, \mathbf{x}) + u$ .

The treatment model is

$$\tau = m_0(\mathbf{z}) + v \quad E(v|\mathbf{z}) = 0$$

where  $\mathbf{z}$  are potentially high-dimensional covariates in the treatment model.  $m_0(\mathbf{z})$  is the expected value of  $\tau$  given  $\mathbf{z}$ . In other words,  $m_0(\mathbf{z})$  is the probability of a unit getting treated given  $\mathbf{z}$ .

The parameter of interest  $\theta_0$  is ATE, ATET, or POMs.

ATE is

$$\theta_0 = E \{g_0(1, \mathbf{x}) - g_0(0, \mathbf{x})\}$$

ATET is

$$\theta_0 = E \{g_0(1, \mathbf{x}) - g_0(0, \mathbf{x})|\tau = 1\}$$

POMs when  $\tau = 1$  is

$$\theta_0 = E \{g_0(1, \mathbf{x})\}$$

POMs when  $\tau = 0$  is

$$\theta_0 = E \{g_0(0, \mathbf{x})\}$$

We have several remarks about the above model.

1. The outcome model can be one of **linear**, **logit**, **probit**, or **poisson**. Let  $\beta_t$  be the outcome model parameters when treatment  $\tau = t$ . The table below provides details about the available functional form of  $g_0(\tau, \mathbf{x})$ .

Outcome model	Functional form for $g_0(\tau = t, \mathbf{x})$
linear	$\mathbf{x}'\beta_t$
logit	$\exp(\mathbf{x}'\beta_t)/\{1 + \exp(\mathbf{x}'\beta_t)\}$
probit	$\Phi(\mathbf{x}'\beta_t)$
poisson	$\exp(\mathbf{x}'\beta_t)$

2. The treatment model can be either **logit** or **probit**. Let  $\gamma$  be the parameters in the treatment model. The table below provides details about the available functional form of  $m_0(\mathbf{z})$ .

Treatment model	Functional form for $m_0(\mathbf{z})$
logit	$\exp(\mathbf{z}'\gamma)/\{1 + \exp(\mathbf{z}'\gamma)\}$
probit	$\Phi(\mathbf{z}'\gamma)$

3. Both  $\mathbf{x}$  and  $\mathbf{z}$  can be high dimensional, and there may be more variables than the number of observations. However, the outcome model parameter  $\beta_t$  and the treatment model parameter  $\gamma$  are assumed to be sparse. This means that there are only a few nonzero elements in both  $\beta_t$  and  $\gamma$ .
4. Although we assume that the functional forms of  $g_0(\cdot)$  and  $m_0(\cdot)$  are known, we do not know which variables should enter the model. We use lasso techniques to select variables from the potential high-dimensional controls  $\mathbf{x}$  and  $\mathbf{z}$ . The resulting estimators should guard against possible model selection errors made by lassos.

Next we discuss the methods that `telasso` uses to estimate ATEs, ATETs, and POMs.

## Neyman orthogonal moments

Denote  $\theta_0$  as the parameter of interest, which can be ATE, ATET, or POM. Following [Chernozhukov et al. \(2018\)](#), we use the following moment conditions to estimate  $\theta_0$ .

The moment condition for estimating ATE is

$$\psi(\mathbf{w}; \theta, \boldsymbol{\eta}) = \{g(1, \mathbf{x}) - g(0, \mathbf{x})\} + \frac{\tau \{y - g(1, \mathbf{x})\}}{m(\mathbf{z})} - \frac{(1 - \tau) \{y - g(0, \mathbf{x})\}}{1 - m(\mathbf{z})} - \theta \quad (1)$$

where  $\mathbf{w} = (y, \mathbf{x}', \mathbf{z}', \tau)'$  and  $\boldsymbol{\eta}$  is the nuisance parameter consisting of  $g(\tau, \mathbf{x})$  and  $m(\mathbf{z})$ . Using the variables selected by lassos,  $g(1, \mathbf{x})$ ,  $g(0, \mathbf{x})$ , and  $m(\mathbf{z})$  are approximations to the true functions  $g_0(1, \mathbf{x})$ ,  $g_0(0, \mathbf{x})$ , and  $m_0(\mathbf{z})$ , respectively.

The moment condition for estimating ATET is

$$\psi(\mathbf{w}; \theta; \boldsymbol{\eta}) = \frac{\tau \{y - g(0, \mathbf{x})\}}{p} - \frac{m(\mathbf{z})(1 - \tau) \{y - g(0, \mathbf{x})\}}{p \{1 - m(\mathbf{z})\}} - \frac{\tau \theta}{p} \quad (2)$$

where  $p = E(\tau)$ .

The moment condition for estimating POM for  $\tau = 1$  is

$$\psi(\mathbf{w}; \theta, \boldsymbol{\eta}) = g(1, \mathbf{x}) + \frac{\tau \{y - g(1, \mathbf{x})\}}{m(\mathbf{z})} - \theta \quad (3)$$

The moment condition for estimating POM for  $\tau = 0$  is

$$\psi(\mathbf{w}; \theta, \boldsymbol{\eta}) = g(0, \mathbf{x}) + \frac{(1 - \tau) \{y - g(0, \mathbf{x})\}}{1 - m(\mathbf{z})} - \theta \quad (4)$$

To estimate  $\theta_0$ , we first use lasso techniques to select variables in the outcome and treatment models. Based on the selected variables, we can estimate the post-lasso predictions for  $g(\tau, \mathbf{x})$  and  $m(\mathbf{z})$ . Denote  $\tilde{\boldsymbol{\eta}}$  as the estimates for  $g(\tau, \mathbf{x})$  and  $m(\mathbf{z})$ .

The estimator  $\hat{\theta}$  for  $\theta_0$  is the solution to

$$\frac{1}{n} \sum_{i=1}^n [\psi_i(\mathbf{w}; \theta, \tilde{\boldsymbol{\eta}})] = 0 \quad (5)$$

The variance estimator for  $\hat{\theta}$  is

$$\frac{1}{n^2} \sum_{i=1}^n [\psi_i(\mathbf{w}; \hat{\theta}, \tilde{\boldsymbol{\eta}})^2] = 0$$

To identify  $\theta_0$ , we need to assume the conditional mean independence, overlap, and independent treatment assignment. For a detailed discussion of these assumptions, see [\[TE\] teffects intro](#) or [\[TE\] teffects intro advanced](#).

The moment conditions defined in (1) to (4) are Neyman orthogonal. Intuitively, Neyman orthogonal moments mean that the estimator or parameter of interest is still consistent even if model selection makes some mild mistakes. For a formal introduction on Neyman orthogonality, see section 2.1 in [Chernozhukov et al. \(2018\)](#).

The moment conditions in (1) to (4) also imply the AIPW estimator for treatment effects. The AIPW estimator is doubly robust in the sense that only one of the outcome model or the treatment model is required to be correctly specified.

## Double machine learning

`telasso` also implements the double machine learning estimator in Chernozhukov et al. (2018) for ATEs, ATETs, and POMs. One advantage of double machine learning is that it allows `telasso` to select more variables in the models. The sparsity assumption is crucial for the validity of the estimators implemented in `telasso`. In other words, the sparsity assumption allows only a few variables to be selected. Double machine learning techniques relax this sparsity requirement to some extent such that more variables can be selected and the estimation results are still valid.

`telasso` with the `xfold()` option implements cross-fit estimation for ATEs, ATETs, and POMs. Cross-fit estimation fits the nuisance parameters and parameter of interest in different samples. It has the following structure.

Let  $K$  be the specified number of cross-fitting folds.

1. Randomly partition the sample into  $K$  subsamples called folds.
2. Define  $I_k$  to be the observations in fold  $k$ , and define  $IC_k$  to be the sample observations not in fold  $k$ .
3. For each  $k = 1, \dots, K$ , fill in the observations of  $i \in I_k$  for the moment condition  $\psi(\cdot)$  in (1), (2), (3), or (4) depending on the specified statistic.
  - a. Using observations  $i \in IC_k$ , perform model selection, and estimate the parameters in the outcome and treatment models.
  - b. Based on the estimates in step 3a, predict the nuisance parameter  $\tilde{\eta}_k$  for observations  $i \in I_k$ .
  - c. Based on  $\tilde{\eta}_k$ , fill in the moment conditions  $\psi(\cdot)$  for observations  $i \in I_k$ .
4. The estimator  $\hat{\theta}$  is the solution in (5).

## Resampling the partitions

The  $K$  folds are chosen once by default. Specify the `resample(#)` option to have the  $K$  folds randomly selected  $\#$  times. This resampling removes the dependence of the estimator on any specifically selected folds, at the cost of more computer time.

Let  $S$  be the specified number of resamples.

1. For each random partition  $s = 1, \dots, S$ , use a cross-fit estimator to obtain point estimate  $\hat{\theta}_s$  and the estimated VCE  $\widehat{\text{Var}}(\hat{\theta}_s)$ .
2. The mean resampling-corrected point estimates are

$$\tilde{\theta} = \frac{1}{S} \sum_{s=1}^S \hat{\theta}_s$$

3. The mean resampling-corrected estimate of the VCE is

$$\widetilde{\text{Var}}(\tilde{\theta}) = \frac{1}{S} \sum_{s=1}^S \left\{ \widehat{\text{Var}}(\hat{\theta}_s) + (\hat{\theta}_s - \tilde{\theta})(\hat{\theta}_s - \tilde{\theta})' \right\}$$

## References

- Chernozhukov, V., D. Chetverikov, M. Demirer, E. Duflo, C. B. Hansen, W. K. Newey, and J. M. Robins. 2018. Double/debiased machine learning for treatment and structural parameters. *Econometrics Journal* 21: C1–C68. <https://doi.org/10.1111/ectj.12097>.
- Chernozhukov, V., and C. B. Hansen. 2004. The effects of 401(k) participation on the wealth distribution: An instrumental quantile regression analysis. *Review of Economics and Statistics* 86: 735–751. <https://doi.org/10.1162/0034653041811734>.
- Farrell, M. H. 2015. Robust inference on average treatment effects with possibly more covariates than observations. *Journal of Econometrics* 189: 1–23. <https://doi.org/10.1016/j.jeconom.2015.06.017>.
- Koch, B., D. M. Vock, and J. Wolfson. 2018. Covariate selection with group lasso and doubly robust estimation of causal effects. *Biometrics* 74: 8–17. <https://doi.org/10.1111/biom.12736>.
- Leeb, H., and B. M. Pötscher. 2005. Model selection and inference: Facts and fiction. *Econometric Theory* 21: 21–59. <https://doi.org/10.1017/S0266466605050036>.
- . 2006. Can one estimate the conditional distribution of post-model-selection estimators? *Annals of Statistics* 34: 2554–2591. <https://doi.org/10.1214/009053606000000821>.
- Poterba, J. M., S. F. Venti, and D. A. Wise. 1994. 401(k) plans and tax-deferred saving. In *Studies in the Economics of Aging*, ed. D. A. Wise, 105–142. New York: National Bureau of Economic Research.
- . 1995. Do 401(k) contributions crowd out other personal saving? *Journal of Public Economics* 58: 1–32. [https://doi.org/10.1016/0047-2727\(94\)01462-W](https://doi.org/10.1016/0047-2727(94)01462-W).
- Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55. <https://doi.org/10.2307/2335942>.

## Also see

- [TE] **telasso postestimation** — Postestimation tools for telasso
- [TE] **teffects aipw** — Augmented inverse-probability weighting
- [LASSO] **Lasso inference intro** — Introduction to inferential lasso models
- [U] **20 Estimation and postestimation commands**

Postestimation commands

[predict](#)

[Remarks and examples](#)

[Also see](#)

## Postestimation commands

The following postestimation commands are of special interest after `telasso`:

Command	Description
<code>teoverlap</code>	overlap plots
<code>tebalance</code>	check balance of covariates
<code>bicplot</code>	plot Bayesian information criterion function
* <code>coefpath</code>	plot path of coefficients
* <code>cvplot</code>	plot cross-validation function
<code>lassocoef</code>	display selected coefficients
<code>lassoinfo</code>	information about lasso estimation results
<code>lassoknots</code>	knot table of coefficient selection and measures of fit
* <code>lassoselect</code>	select alternative $\lambda^*$

\* `coefpath`, `cvplot`, and `lassoselect` require the selection method of the lasso to be `selection(cv)`, `selection(adaptive)`, or `selection(bic)`. See [\[LASSO\] lasso options](#).

The following standard postestimation commands are also available:

Command	Description
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of coefficients
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
<code>predict</code>	treatment effects, conditional means at treatment, propensity scores, etc.
<code>predictnl</code>	point estimates for generalized predictions
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

# predict

## Description for predict

`predict` creates a new variable containing predictions such as treatment effects, potential outcomes, conditional means, propensity scores, and linear predictions.

## Menu for predict

Statistics > Postestimation

## Syntax for predict

```
predict [type] { stub* | newvar | newvarlist } [if] [in]
      [, statistic tlevel(treat_level)]
```

<i>statistic</i>	Description
Main	
<code>te</code>	treatment effect; the default
<code>cmean</code>	conditional mean at treatment level
<code>ps</code>	propensity score
<code>xb</code>	linear prediction
<code>psxb</code>	linear prediction for propensity score

Option `tlevel()` may not be combined with `te` or `psxb`.

If you do not specify `tlevel()` and only specify one new variable, then `cmean`, `ps`, and `xb` assume `tlevel()` specifies the control.

You specify one or two new variables with `cmean`, `ps`, and `xb`.

You specify one new variable with `te` and `psxb`.

## Options for predict

### Main

`te`, the default, calculates the treatment effect for the noncontrol treatment level. You need to specify only one new variable.

`cmean` calculates the conditional mean for each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify two new variables corresponding to the control and noncontrol treatment levels.

`ps` calculates the propensity score of each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify two new variables corresponding to the control and noncontrol treatment levels.

`xb` calculates the linear prediction at each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify two new variables corresponding to the control and noncontrol treatment levels.



`psxb` calculates the linear prediction for the propensity score at the noncontrol level of the treatment.

You need to specify only one new variable.

`tlevel(treat_level)` specifies the treatment level for prediction.

## Remarks and examples

Some of the `telasso` postestimation commands explore the lasso results computed within `telasso`. Here is a list of such commands: `coefpath`, `cvplot`, `bicplot`, `lassoknots`, `lassoselect`, and `lassocoef`.

When referring to a lasso result computed by `telasso`, there is a distinction between the outcome model and the treatment model. To refer to the lasso result for the treatment model, we need to specify the treatment variable with the `for(tvar)` option. In contrast, to refer to the lasso result for the outcome model, we need to specify the outcome variable at a specific treatment level with the `for(ovar)` and `tlevel(#)` options. In summary, for the treatment model, the `for(tvar)` option is required; for the outcome model, both the `for(ovar)` and the `tlevel(#)` options are required.

Examples that demonstrate how to use the `telasso` command and explore the lasso results using the postestimation tools can be found in *Remarks and examples* in [TE] `telasso`.

## Also see

[TE] `telasso` — Treatment-effects estimation using lasso

[U] 20 Estimation and postestimation commands

# Title

**teoverlap** — Overlap plots

[Description](#)  
[Options](#)  
[Also see](#)

[Quick start](#)  
[Remarks and examples](#)

[Menu](#)  
[Stored results](#)

[Syntax](#)  
[References](#)

## Description

One of the assumptions required to use the `teffects` and `stteffects` estimators is the overlap assumption, which states that each individual has a positive probability of receiving each treatment level. `teoverlap`, a postestimation command, plots the estimated densities of the probability of getting each treatment level. These plots can be used to check whether the overlap assumption is violated.

## Quick start

Visually check whether the overlap assumption is violated

```
teoverlap
```

As above, but use the Epanechnikov kernel function

```
teoverlap, kernel(epanechnikov)
```

Plot probability of getting treatment 3 for subjects receiving treatments 2 or 3 of a multivalued treatment

```
teoverlap, ptlevel(3) tlevels(2 3)
```

As above, and change legend labels to “Treated 2” and “Treated 3”

```
teoverlap, ptlevel(3) tlevels(2 3) ///  
    legend(label(1 "Treated 2") label(2 "Treated 3"))
```

## Menu

Statistics > Treatment effects > Overlap plots

## Syntax

```
teoverlap [ , treat_options kden_options ]
```

<i>treat_options</i>	Description
Main	
<u>pt</u> level( <i>treat_level</i> )	calculate predicted probabilities for treatment level <i>treat_level</i> ; by default, <code>ptlevel()</code> corresponds to the first treatment level
<u>t</u> levels( <i>treatments</i> )	specify conditioning treatment levels; default is all treatment levels
<u>n</u> olabel	use treatment level values and not value labels in legend and axis titles

<i>kden_options</i>	Description
Main	
<u>k</u> ernel( <i>kernel</i> )	specify kernel function; default is <code>kernel(triangle)</code>
<u>n</u> (#)	estimate densities using # points; default is $e(N)$ , the number of observations in the estimation sample
<u>b</u> width(#)	half-width of kernel
<u>a</u> t( <i>var_x</i> )	estimate densities using the values specified by <i>var_x</i>
Kernel plots	
<u>l</u> ine# <u>o</u> pts( <i>cline_options</i> )	affect rendition of density for conditioning treatment #
Add plots	
<u>a</u> ddplot( <i>plot</i> )	add other plots to the generated graph
Y axis, X axis, Titles, Legend, Overall	
<i>twoway_options</i>	any options other than <code>by()</code> documented in [G-3] <i>twoway_options</i>

<i>kernel</i>	Description
<u>t</u> riangle	triangle kernel function; the default
<u>e</u> panechnikov	Epanechnikov kernel function
<u>e</u> pan2	alternative Epanechnikov kernel function
<u>b</u> iweight	biweight kernel function
<u>c</u> osine	cosine trace kernel function
<u>g</u> aussian	Gaussian kernel function
<u>p</u> arzen	Parzen kernel function
<u>r</u> ectangle	rectangle kernel function

collect is allowed; see [U] 11.1.10 Prefix commands.

## Options

Main

`ptlevel(treat_level)` specifies that predicted probabilities be calculated for treatment level *treat\_level*. The default is `ptlevel(first)`, where *first* is the first treatment level.

`tlevels(treatments)` specifies the observations for which to obtain predicted probabilities. By default, all treatment levels are used. Specify *treatments* as a space-delimited list.

For instance,

```
. teoverlap, ptlevel(1) tlevels(1 2)
```

says to predict the probability of getting treatment level 1 for those subjects who actually obtained treatment levels 1 or 2.

`nolabel` specifies that treatment level values and not value labels be used in legend and axis titles.

`kernel(kernel)` specifies the kernel function for use in calculating the kernel density estimates. The default kernel is the triangle kernel (`triangle`).

`n(#)` specifies the number of points at which the density estimate is to be evaluated. The default is `e(N)`, the estimation sample size.

`bwidth(#)` specifies the half-width of the kernel, the width of the density window around each point. If `bwidth()` is not specified, the “optimal” width is calculated and used; see [R] [kdensity](#). The optimal width is the width that would minimize the mean integrated squared error if the data were Gaussian and a Gaussian kernel were used, so it is not optimal in any global sense. In fact, for multimodal and highly skewed densities, this width is usually too wide and oversmooths the density ([Silverman 1986](#)).

`at(var_x)` specifies a variable that contains the values at which the density should be estimated. This option allows you to more easily obtain density estimates for different variables or different subsamples of a variable and then overlay the estimated densities for comparison.

#### Kernel plots

`line#opts(cline_options)` affect the rendition of the plotted kernel density estimates. See [G-3] [cline\\_options](#).

#### Add plots

`addplot(plot)` provides a way to add other plots to the generated graph. See [G-3] [addplot\\_option](#).

#### Y axis, X axis, Titles, Legend, Overall

*twoway\_options* are any of the options documented in [G-3] [twoway\\_options](#), excluding `by()`. These include options for titling the graph (see [G-3] [title\\_options](#)) and for saving the graph to disk (see [G-3] [saving\\_option](#)).

## Remarks and examples

`teoverlap` plots the estimated densities of the probability of getting each treatment level after `teffects`.

These plots can be used to check whether the overlap assumption is violated. The overlap assumption is satisfied when there is a chance of seeing observations in both the control and the treatment groups at each combination of covariate values; see [TE] [teffects intro](#) or [TE] [teffects intro advanced](#).

The overlap assumption is required by the estimators implemented in `teffects`. Intuitively, when the overlap assumption is violated, we cannot predict, or otherwise account for, the unobserved outcomes for some individuals.

There is evidence that the overlap assumption is violated when an estimated density has too much mass around 0 or 1; see [Busso, DiNardo, and McCrary \(2014\)](#). An implication of this point is that when the overlap assumption is violated, the estimated densities will have relatively little mass in the regions in which they overlap.

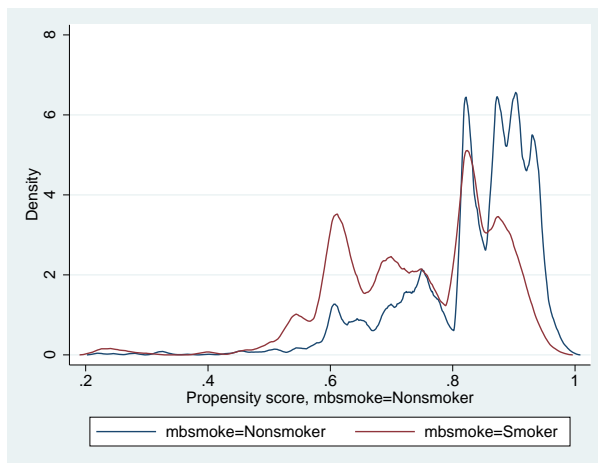
### ▷ Example 1: Assumption not violated

Continuing with [example 1](#) of [TE] `teffects ipw`, we estimate the average treatment effect of smoking on birthweight and then draw the overlap plot:

```
. use https://www.stata-press.com/data/r17/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ipw (bweight) (mbsmoke mmarried c.mage#c.mage fbaby medu, probit)
Iteration 0:  EE criterion = 4.621e-21
Iteration 1:  EE criterion = 7.358e-26
Treatment-effects estimation      Number of obs      =      4,642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: probit
```

bweight		Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.					
ATE							
mbsmoke (Smoker vs Nonsmoker)	-230.6886	25.81524	-8.94	0.000	-281.2856	-180.0917	
POmean							
mbsmoke Nonsmoker	3403.463	9.571369	355.59	0.000	3384.703	3422.222	

```
. teoverlap
```



The graph displays the estimated density of the predicted probabilities that a nonsmoking mother is a nonsmoker and the estimated density of the predicted probabilities that a smoking mother is a nonsmoker.

Neither plot indicates too much probability mass near 0 or 1, and the two estimated densities have most of their respective masses in regions in which they overlap each other. Thus there is no evidence that the overlap assumption is violated.

◀

### ► Example 2: Assumption violated

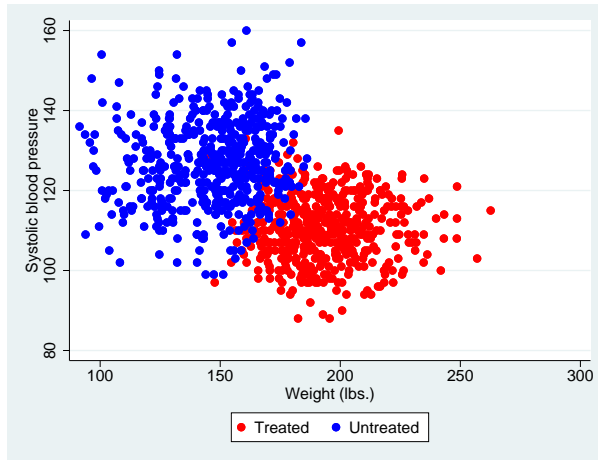
This example produces an overlap plot that indicates a failure of the overlap assumption. We will use simulated data, so we know that the assumption is not true.

In our simulated dataset, some of the 1,000 adult males were given drug XY1 for high blood pressure and others were not. A scatterplot of systolic blood pressure (`systolic`) and weight (`weight`) reveals that heavier men were given the treatment. (The scatterplots corresponding to the treatment group are colored red, while the scatterplots corresponding to the control group are colored blue.)

```

. use https://www.stata-press.com/data/r17/systolic2
(Systolic blood pressure)
. twoway (scatter systolic weight if xy1==1, mcolor(red))
>       (scatter systolic weight if xy1==0, mcolor(blue)),
>       legend(label(1 "Treated") label(2 "Untreated"))

```



There are no observations in the treated group for small weights, and there are no observations in the control group for large weights. There is clear evidence that the overlap assumption is violated.

Drawing an overlaid scatterplot is a straightforward way to check the overlap assumption in this example because there is only one covariate. This method is not available when there is more than one covariate. The predicted probability is a one-dimensional measure that captures the relevant multivariate information.

Below we estimate the parameters needed to calculate the predicted probabilities. The `pstolerance(1e-8)` option is specified to ensure that estimation is performed as long as the predicted probabilities are at least as large as  $1e-8$ .

```

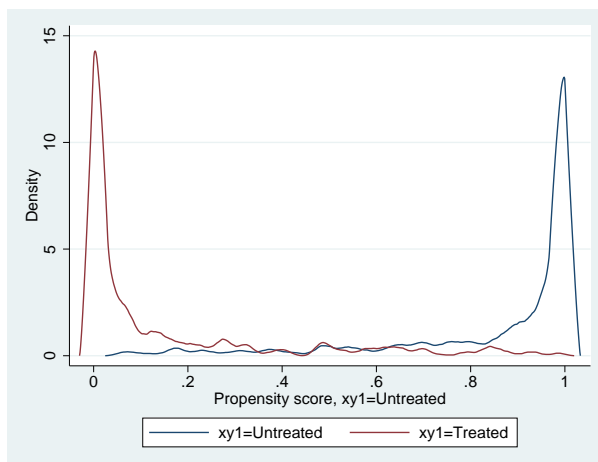
. teffects ipw (systolic) (xy1 weight), pstolerance(1e-8)
Iteration 0: EE criterion = 9.523e-18
Iteration 1: EE criterion = 3.489e-28
Treatment-effects estimation      Number of obs      =      1,000
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: logit

```

systolic	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
xy1 (Treated vs Untreated)	-16.23679	2.191703	-7.41	0.000	-20.53245	-11.94114
P0mean						
xy1 Untreated	127.9094	.7004533	182.61	0.000	126.5365	129.2822

Now we can obtain the overlap plot.

```
. teoverlap
```



The estimated density of the predicted probabilities that a treated individual is not assigned to XY1 treatment has most of its mass near 0. The estimated density of the predicted probabilities that an untreated individual is not assigned to XY1 treatment has most of its mass near 1. Note that the two have very little mass in the region in which they overlap. There is clear evidence that the overlap assumption is violated.

◀

## Stored results

`teoverlap` stores the following in `r()`:

### Scalars

<code>r(bwidth<math>j</math>)</code>	kernel bandwidth for treatment level $j$
<code>r(n<math>j</math>)</code>	number of points at which the estimate was evaluated for treatment level $j$
<code>r(scale<math>j</math>)</code>	density bin width for treatment level $j$

### Macros

<code>r(kernel)</code>	name of kernel
------------------------	----------------

## References

Busso, M., J. DiNardo, and J. McCrary. 2014. New evidence on the finite sample properties of propensity score reweighting and matching estimators. *Review of Economics and Statistics* 96: 885–897. [https://doi.org/10.1162/REST\\_a\\_00431](https://doi.org/10.1162/REST_a_00431).

Silverman, B. W. 1986. *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall.



## Also see

- [TE] [stteffects](#) — Treatment-effects estimation for observational survival-time data
- [TE] [stteffects ipw](#) — Survival-time inverse-probability weighting
- [TE] [stteffects ipwra](#) — Survival-time inverse-probability-weighted regression adjustment
- [TE] [teffects](#) — Treatment-effects estimation for observational data
- [TE] [teffects aipw](#) — Augmented inverse-probability weighting
- [TE] [teffects ipw](#) — Inverse-probability weighting
- [TE] [teffects ipwra](#) — Inverse-probability-weighted regression adjustment
- [TE] [teffects nnmatch](#) — Nearest-neighbor matching
- [TE] [teffects psmatch](#) — Propensity-score matching
- [TE] [teffects ra](#) — Regression adjustment
- [TE] [telasso](#) — Treatment-effects estimation using lasso

# Glossary

**AIPW estimator.** See *augmented inverse-probability-weighted estimator*.

**analysis time.** Analysis time is like time, except that 0 has a special meaning:  $t = 0$  is the time of onset of risk, the time when failure first became possible.

Analysis time is usually not what is recorded in a dataset. A dataset of patients might record calendar time. Calendar time must then be mapped to analysis time.

The letter  $t$  is reserved for time in analysis-time units. The term *time* is used for time measured in other units.

The *origin* is the *time* corresponding to  $t = 0$ , which can vary subject to subject. Thus  $t = \text{time} - \text{origin}$ .

**ATE.** See *average treatment effect*.

**ATET.** See *average treatment effect on the treated*.

**augmented inverse-probability-weighted estimator.** An augmented inverse-probability-weighted (AIPW) estimator is an inverse-probability-weighted estimator that includes an augmentation term that corrects the estimator when the treatment model is misspecified. When the treatment is correctly specified, the augmentation term vanishes as the sample size becomes large. An AIPW estimator uses both an outcome model and a treatment model and is a doubly robust estimator.

**average treatment effect.** The average treatment effect is the average effect of the treatment among all individuals in a population.

**average treatment effect on the treated.** The average treatment effect on the treated is the average effect of the treatment among those individuals who actually get the treatment.

**censored, left-censored, and right-censored.** An observation is left-censored when the exact time of failure is not known; it is merely known that the failure occurred before  $t_l$ . Suppose that the event of interest is becoming employed. If a subject is already employed when first interviewed, his outcome is left-censored.

An observation is right-censored when the time of failure is not known; it is merely known that the failure occurred after  $t_r$ . If a patient survives until the end of a study, the patient's time of death is right-censored.

In common usage, *censored* without a modifier means right-censored.

Also see *truncation, left-truncation, and right-truncation*.

**CI assumption.** See *conditional-independence assumption*.

**conditional mean.** The conditional mean expresses the average of one variable as a function of some other variables. More formally, the mean of  $y$  conditional on  $\mathbf{x}$  is the mean of  $y$  for given values of  $\mathbf{x}$ ; in other words, it is  $E(y|\mathbf{x})$ .

A conditional mean is also known as a regression or as a conditional expectation.

**conditional-independence assumption.** The conditional-independence assumption requires that the common variables that affect treatment assignment and treatment-specific outcomes be observable. The dependence between treatment assignment and treatment-specific outcomes can be removed by conditioning on these observable variables.

This assumption is also known as a selection-on-observables assumption because its central tenet is the observability of the common variables that generate the dependence.

**counterfactual.** A counterfactual is an outcome a subject would have obtained had that subject received a different level of treatment. In the binary-treatment case, the counterfactual outcome for a person who received treatment is the outcome that person would have obtained had the person instead not received treatment; similarly, the counterfactual outcome for a person who did not receive treatment is the outcome that person would have obtained had the person received treatment.

Also see *potential outcome*.

**doubly robust estimator.** A doubly robust estimator only needs one of two auxiliary models to be correctly specified to estimate a parameter of interest.

Doubly robust estimators for treatment effects are consistent when either the outcome model or the treatment model is correctly specified.

**EE estimator.** See *estimating-equation estimator*.

**estimating-equation estimator.** An estimating-equation (EE) estimator calculates parameters estimates by solving a system of equations. Each equation in this system is the sample average of a function that has mean zero.

These estimators are also known as  $M$  estimators or  $Z$  estimators in the statistics literature and as generalized method of moments (GMM) estimators in the econometrics literature.

**failure event.** Survival analysis is really time-to-failure analysis, and the failure event is the event under analysis. The failure event can be death, heart attack, myopia, or finding employment. Many authors—including Stata—write as if the failure event can occur only once per subject, but when we do, we are being sloppy. Survival analysis encompasses repeated failures, and all of Stata's survival analysis features can be used with repeated-failure data.

**hazard, cumulative hazard, and hazard ratio.** The hazard or hazard rate at time  $t$ ,  $h(t)$ , is the instantaneous rate of failure at time  $t$  conditional on survival until time  $t$ . Hazard rates can exceed 1. Say that the hazard rate were 3. If an individual faced a constant hazard of 3 over a unit interval and if the failure event could be repeated, the individual would be expected to experience three failures during the time span.

The cumulative hazard,  $H(t)$ , is the integral of the hazard function  $h(t)$ , from 0 (the onset of risk) to  $t$ . It is the total number of failures that would be expected to occur up until time  $t$ , if the failure event could be repeated. The relationship between the cumulative hazard function,  $H(t)$ , and the survivor function,  $S(t)$ , is

$$S(t) = \exp\{-H(t)\}$$

$$H(t) = -\ln\{S(t)\}$$

The hazard ratio is the ratio of the hazard function evaluated at two different values of the covariates:  $h(t | \mathbf{x})/h(t | \mathbf{x}_0)$ . The hazard ratio is often called the relative hazard, especially when  $h(t | \mathbf{x}_0)$  is the baseline hazard function.

**i.i.d. sampling assumption.** See *independent and identically distributed sampling assumption*.

**independent and identically distributed sampling assumption.** The independent and identically distributed (i.i.d.) sampling assumption specifies that each observation is unrelated to (independent of) all the other observations and that each observation is a draw from the same (identical) distribution.

**individual-level treatment effect.** An individual-level treatment effect is the difference in an individual's outcome that would occur because this individual is given one treatment instead of another. In other words, an individual-level treatment effect is the difference between two potential outcomes for an individual.

For example, the blood pressure an individual would obtain after taking a pill minus the blood pressure an individual would obtain had that person not taken the pill is the individual-level treatment effect of the pill on blood pressure.

**inverse-probability-weighted estimators.** Inverse-probability-weighted (IPW) estimators use weighted averages of the observed outcome variable to estimate the potential-outcome means. The weights are the reciprocals of the treatment probabilities estimated by a treatment model.

**inverse-probability-weighted regression-adjustment estimators.**

Inverse-probability-weighted regression-adjustment (IPWRA) estimators use the reciprocals of the estimated treatment probability as weights to estimate missing-data-corrected regression coefficients that are subsequently used to compute the potential-outcome means.

**IPW estimators.** See *inverse-probability-weighted estimators*.

**IPWRA estimators.** See *inverse-probability-weighted regression-adjustment estimators*.

**left-censored.** See *censored, left-censored, and right-censored*.

**left-truncation.** See *truncation, left-truncation, and right-truncation*.

**matching estimator.** An estimator that compares differences between the outcomes of similar—that is, matched—individuals. Each individual that receives a treatment is matched to a similar individual that does not get the treatment, and the difference in their outcomes is used to estimate the individual-level treatment effect. Likewise, each individual that does not receive a treatment is matched to a similar individual that does get the treatment, and the difference in their outcomes is used to estimate the individual-level treatment effect.

**multiple-record st data.** See *st data*.

**multivalued treatment effect.** A multivalued treatment refers to a treatment that has more than two values. For example, a person could have taken a 20 mg dose of a drug, a 40 mg dose of the drug, or not taken the drug at all.

**nearest-neighbor matching.** Nearest-neighbor matching uses the distance between observed variables to find similar individuals.

**observational data.** In observational data, treatment assignment is not controlled by those who collected the data; thus some common variables affect treatment assignment and treatment-specific outcomes.

**outcome model.** An outcome model is a model used to predict the outcome as a function of covariates and parameters.

**overlap assumption.** The overlap assumption requires that each individual have a positive probability of each possible treatment level.

**POMs.** See *potential-outcome means*.

**potential outcome.** The potential outcome is the outcome an individual would obtain if given a specific treatment.

For example, an individual has one potential blood pressure after taking a pill and another potential blood pressure had that person not taken the pill.

**potential-outcome means.** The potential-outcome means refers to the means of the potential outcomes for a specific treatment level.

The mean blood pressure if everyone takes a pill and the mean blood pressure if no one takes a pill are two examples.

The average treatment effect is the difference between potential-outcome mean for the treated and the potential-outcome mean for the not treated.

**propensity score.** The propensity score is the probability that an individual receives a treatment.

**propensity-score matching.** Propensity-score matching uses the distance between estimated propensity scores to find similar individuals.

**regression-adjustment estimators.** Regression-adjustment estimators use means of predicted outcomes for each treatment level to estimate each potential-outcome mean.

**right-censored.** See *censored, left-censored, and right-censored*.

**right-truncation.** See *truncation, left-truncation, and right-truncation*.

**selection-on-observables.** See *conditional-independence assumption*.

**shape parameter.** A shape parameter governs the shape of a probability distribution. One example is the parameter  $p$  of the Weibull model.

**single-record st data.** See *st data*.

**smooth treatment-effects estimator.** A smooth treatment-effects estimator is a smooth function of the data so that standard methods approximate the distribution of the estimator. The RA, IPW, AIPW, and IPWRA estimators are all smooth treatment-effects estimators while the nearest-neighbor matching estimator and the propensity-score matching estimator are not.

**st data.** st stands for survival time. In survival-time data, each observation represents a span of survival, recorded in variables  $t_0$  and  $t$ . For instance, if in an observation  $t_0$  were 3 and  $t$  were 5, the span would be  $(t_0, t]$ , meaning from just after  $t_0$  up to and including  $t$ .

Sometimes variable  $t_0$  is not recorded;  $t_0$  is then assumed to be 0. In such a dataset, an observation that had  $t = 5$  would record the span  $(0, 5]$ .

Each observation also includes a variable  $d$ , called the failure variable, which contains 0 or nonzero (typically, 1). The failure variable records what happened at the end of the span: 0, the subject was still alive (had not yet failed) or 1, the subject died (failed).

Sometimes variable  $d$  is not recorded;  $d$  is then assumed to be 1. In such a dataset, all time-span observations would be assumed to end in failure.

Finally, each observation in an st dataset can record the entire history of a subject or each can record a part of the history. In the latter case, groups of observations record the full history. One observation might record the period  $(0, 5]$  and the next,  $(5, 8]$ . In such cases, there is a variable ID that records the subject for which the observation records a time span. Such data are called multiple-record st data. When each observation records the entire history of a subject, the data are called single-record st data. In the single-record case, the ID variable is optional.

See [ST] **stset**.

**survival-time data.** See *st data*.

**survivor function.** Also known as the survivorship function and the survival function, the survivor function,  $S(t)$ , is 1) the probability of surviving beyond time  $t$ , or equivalently, 2) the probability that there is no failure event prior to  $t$ , 3) the proportion of the population surviving to time  $t$ , or equivalently, 4) the reverse cumulative distribution function of  $T$ , the time to the failure event:  $S(t) = \Pr(T > t)$ . Also see *hazard, cumulative hazard, and hazard ratio*.

**treatment model.** A treatment model is a model used to predict treatment-assignment probabilities as a function of covariates and parameters.

**truncation, left-truncation, and right-truncation.** In survival analysis, truncation occurs when subjects are observed only if their failure times fall within a certain observational period of a study. Censoring, on the other hand, occurs when subjects are observed for the whole duration of a study, but the exact times of their failures are not known; it is known only that their failures occurred within a certain time span.

Left-truncation occurs when subjects come under observation only if their failure times exceed some time  $t_l$ . It is only because they did not fail before  $t_l$  that we even knew about their existence. Left-truncation differs from left-censoring in that, in the censored case, we know that the subject failed before time  $t_l$ , but we just do not know exactly when.

Imagine a study of patient survival after surgery, where patients cannot enter the sample until they have had a post-surgical test. The patients' survival times will be left-truncated. This is a "delayed entry" problem, one common type of left-truncation.

Right-truncation occurs when subjects come under observation only if their failure times do not exceed some time  $t_r$ . Right-truncated data typically occur in registries. For example, a cancer registry includes only subjects who developed a cancer by a certain time, and thus survival data from this registry will be right-truncated.

**unconfoundedness.** See *conditional-independence assumption*.

**weighted-regression-adjustment estimator.** Weighted-regression-adjustment estimators use means of predicted outcomes for each treatment level to estimate each potential-outcome mean. The weights are used to estimate censoring-adjusted regression coefficients.

# Subject and author index

See the [combined subject index](#) and the [combined author index](#) in the *Stata Index*.