Structural Equation Modeling with categorical variables

Yves Rosseel

Department of Data Analysis

Ghent University

Summer School – Using R for personality research
August 23–28, 2014
Bertinoro, Italy

Contents

1 Structural Equation Modeling with categorical variables 3
1.1 Categorical data analysis 3
1.2 The logistic regression model 6
1.3 The probit regression model 17
1.4 Regression with an ordinal response 23
1.5 SEM with categorical (endogenous) variables: two approaches . . 40
1.6 Multiple group analysis with categorical data 73
1.7 Full information approach: marginal maximum likelihood 91
1.8 PML: pairwise maximum likelihood 94

1 Structural Equation Modeling with categorical variables

1.1 Categorical data analysis

continuous/numerical data

- interval or ratio scale (e.g. income, height, weight, age, reaction time, blood pressure, ...

categorical/discrete data

- limited set of possible outcomes/categories
- nominal or binary: gender, dead/alive, country, race/ethnicity, ...
- ordinal: ses (high, middle, low), age group (young, middle, old), likert scales (agree strongly, agree, neutral, disagree, disagree strongly)
- counts: number of deadly accidents this week

categorical data analysis

- (regression models:) response/dependent variable is a categorical variable
  - probit/logistic regression
  - multinomial regression
  - ordinal logit/probit regression
  - Poisson regression
  - generalized linear (mixed) models
- all (dependent) variables are categorical (contingency tables, loglinear analysis)
- other analyses:
  - exploratory/confirmatory factor analysis with binary/ordered data (Item Response Theory, IRT)
  - structural equation modeling with binary/ordered data
  - ...

endogenous versus exogenous

- the categorical variables are exogenous only
  - for example, ANOVA
  - standard approach: convert to dummy variables (if the categorical variable has K levels, we only need $K - 1$ dummy variables)
  - many functions in R do this automatically (`lm()`, `glm()`, `lme()`, `lmer()`, ... if the categorical variable has been declared as a 'factor')
  - but NOT in lavaan; you have to manually construct the dummy variables yourself (before calling any of the lavaan fitting functions)
  - the same for interaction terms (product terms), quadratic terms, ...
  - binary exogenous variables: should be coded as 0/1 or 1/2
- if the categorical variables are endogenous, we need special methods

1.2 The logistic regression model

- generalized linear model (GLM) with binomial random component and logit link function
- the logistic regression model with 1 (continuous) predictor:

$$\log\left(\frac{\pi(x)}{1-\pi(x)}\right) = \text{logit}[\pi(x)] = \beta_0 + \beta_1 x$$

  where $\pi(x)$ denotes the probability of success $P(y = 1|x)$
- β_1 represents the change –per unit increase in x– in the logit value logit$[\pi(x)]$
- relationship between $\pi(x)$ and x:

$$\pi(x) = \frac{\exp(\beta_0 + \beta_1 x)}{1 + \exp(\beta_0 + \beta_1 x)}$$

  - the sign of β_1 determines whether $\pi(x)$ is increasing or decreasing
  - the rate of change in $\pi(x)$ –per unit increase in x– varies (slower when $\pi(x)$ approaches 0 or 1)

- exponentiating both sides of the model gives:

$$\frac{\pi(x)}{1-\pi(x)} = \exp\left(\beta_0 + \beta_1 x\right)$$

the *odds* increase/decrease multiplicatively by $\exp(\beta_1)$ per unit increase in x; if $(x_2 - x_1) = 1$:

$$\text{odds}(x_2) = \text{odds}(x_1) \times \exp(\beta_1)$$

- $\exp(\beta_1)$ is the *odds ratio* corresponding to the 2×2 table with columns ($y = 1$ and $y = 0$) and rows ($x + 1$ and x):

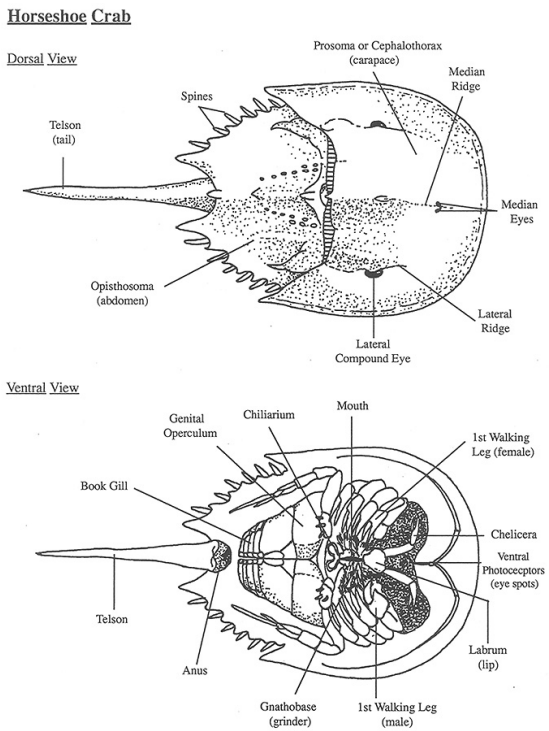
	$y = 1$	$y = 0$
$x + 1$	$\hat{\mu}_{11}$	$\hat{\mu}_{12}$
x	$\hat{\mu}_{21}$	$\hat{\mu}_{22}$

with $\hat{\mu}$'s the expected/fitted frequencies in each cell; the odds ratio for this 2×2 table is $(\hat{\mu}_{11} \times \hat{\mu}_{22})/(\hat{\mu}_{12} \times \hat{\mu}_{21}) = \exp(\beta_1)$

horseshoe crab mating example (binary version)

- 173 horseshoe crabs
- y = female crab has at least one satellite ($y = 1$) or none ($y = 0$)
- x = carapace width (cm)

At least one satellite (y)	Width (x)
1	28.3
1	26.0
0	25.5
0	21.0
1	29.0
...	


Horseshoe Crab
Dorsal View
Prosoma or Cephalothorax (carapace)
Median Ridge
Spines
Telson (tail)
Median Eyes
Opisthosoma (abdomen)
Lateral Ridge
Lateral Compound Eye
Ventral View
Mouth
Genital Operculum
Chiliarium
1st Walking Leg (female)
Book Gill
Chelicera
Ventral Photoceptors (eye spots)
Telson
Labrum (lip)
Anus
Gnathobase (grinder)
1st Walking Leg (male)

R code

```
> Crabs <- read.table("http://www.da.ugent.be/datasets/crab.dat", header=T)
> Crabs$y <- ifelse(Crabs$Sa > 0, 1, 0)
> fit <- glm(y ~ W, data=Crabs, family=binomial)
> summary(fit)

Call:
glm(formula = y ~ W, family = binomial, data = Crabs)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0281  -1.0458   0.5480   0.9066   1.6942

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -12.3508     2.6287  -4.698 2.62e-06 ***
W             0.4972     0.1017   4.887 1.02e-06 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 225.76  on 172  degrees of freedom
Residual deviance: 194.45  on 171  degrees of freedom
AIC: 198.45

Number of Fisher Scoring iterations: 4
```

results logistic regression model

- the model predicts:

$$\text{logit}[\hat{\pi}(x)] = -12.3508 + 0.4972 \times \text{Width}$$

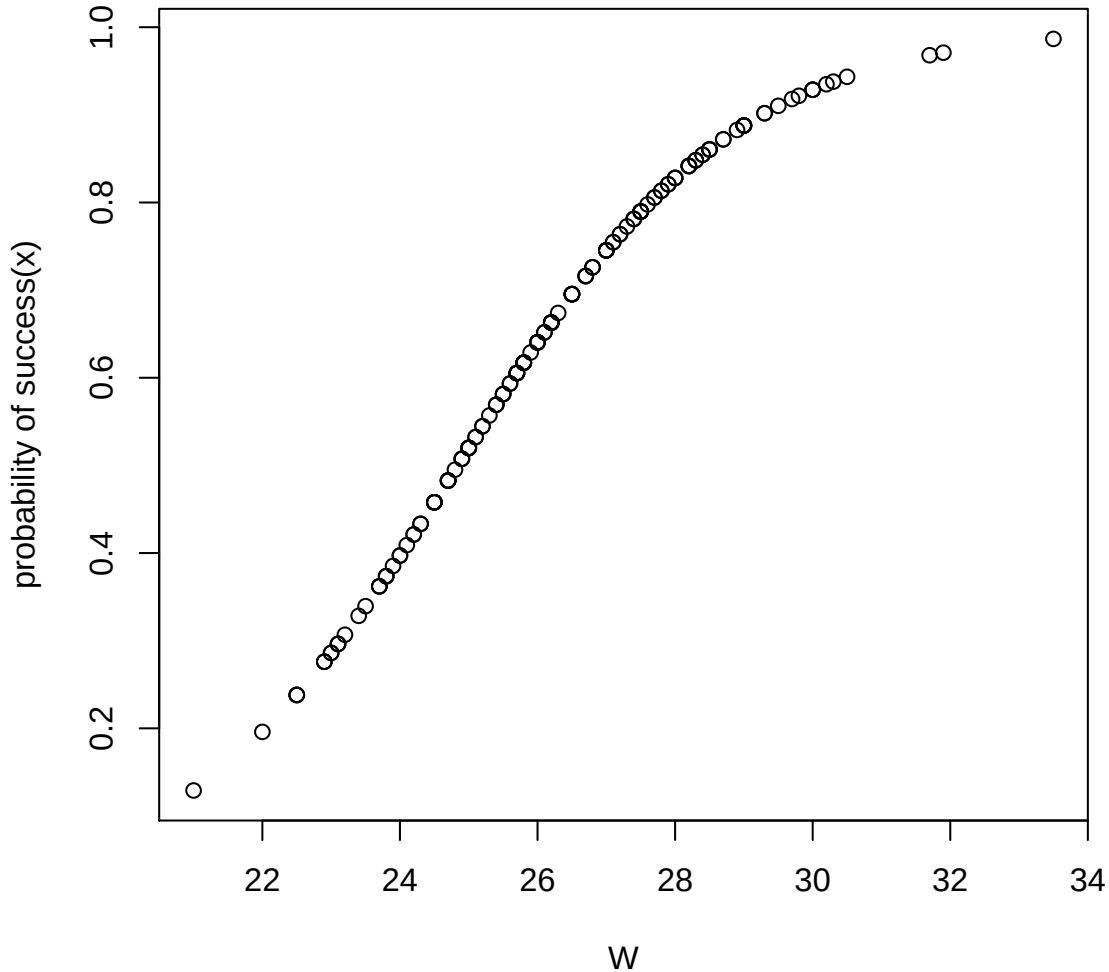
- in terms of the probability success:

$$\hat{\pi}(x) = \frac{\exp(-12.3508 + 0.4972 \times \text{Width})}{1 + \exp(-12.3508 + 0.4972 \times \text{Width})}$$

- the predicted probability success for some values of x (=Width)

$$\hat{\pi}(x = 22) = 0.196$$
$$\hat{\pi}(x = 24) = 0.397$$
$$\hat{\pi}(x = 26) = 0.640$$
$$\hat{\pi}(x = 28) = 0.828$$
$$\hat{\pi}(x = 30) = 0.929$$

plot probability of success as a function of W



- the exponentiated coefficients:

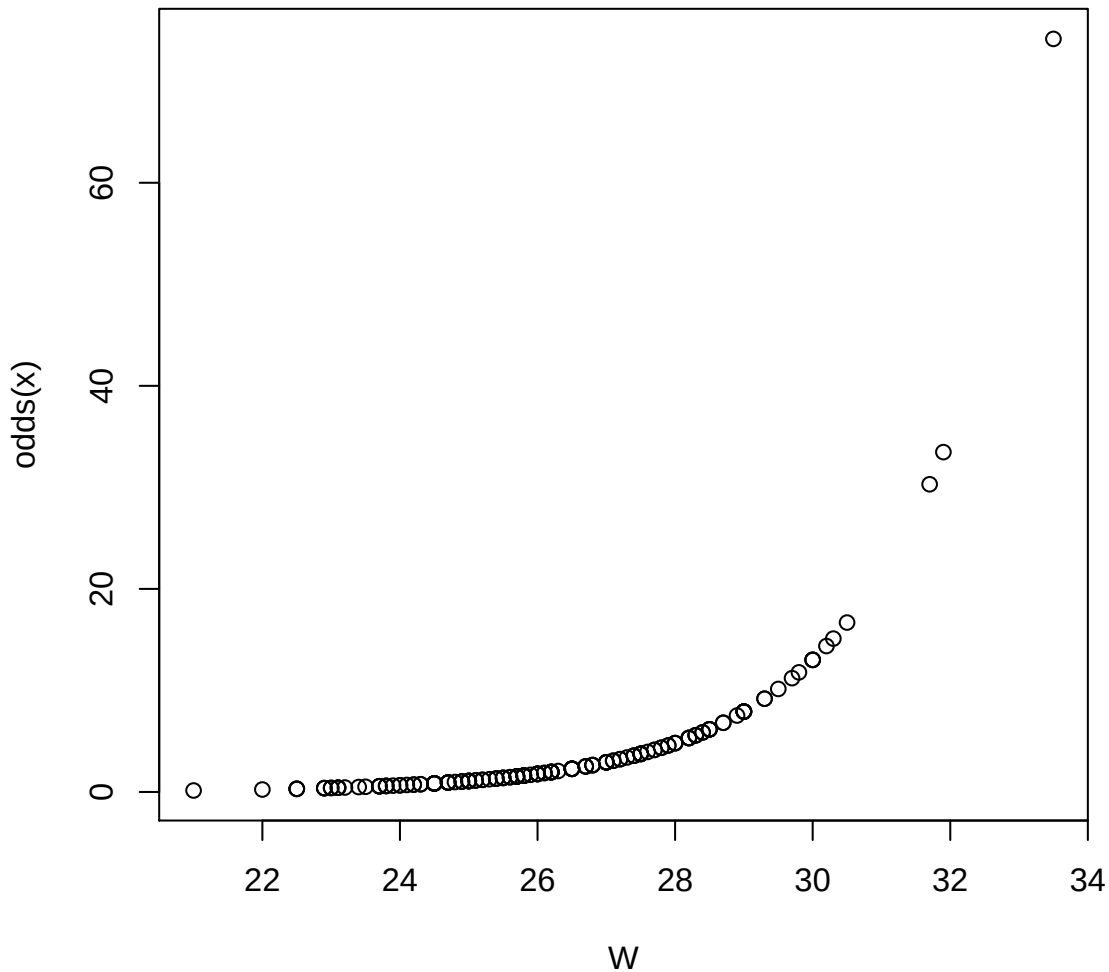
$\exp(\beta_0)$	$\exp(\beta_1)$
0.000	1.644

- interpretation: for a 1-unit change in x, the **odds** increase multiplicatively by 1.644
- the predicted odds for some values of x (=Width)

$$\begin{aligned}
\text{odds}(x = 22) &= 0.244 \\
\text{odds}(x = 24) &= 0.659 \\
\text{odds}(x = 26) &= 1.781 \\
\text{odds}(x = 28) &= 4.815 \\
\text{odds}(x = 30) &= 13.015
\end{aligned}$$

$$\begin{aligned}
\text{odds}(x = 30) &= \text{odds}(x = 28) \times \exp(\beta \times (30 - 28)) \\
13.015 &= 4.815 \times 2.7031
\end{aligned}$$

plot odds as a function of W



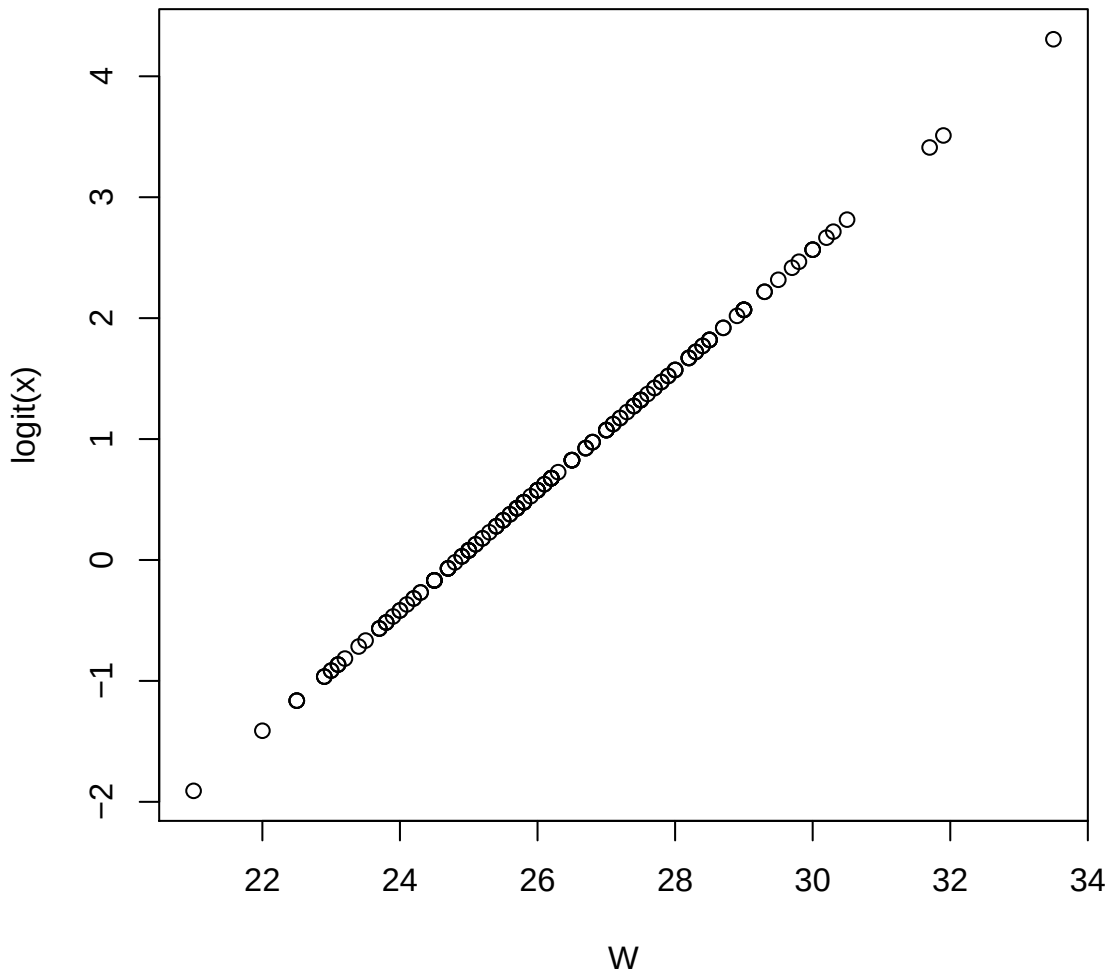
- the relationship between the 'logits' and x is linear:

$$\text{logit}[\hat{\pi}(x)] = -12.3508 + 0.4972 \times \text{Width}$$

- the predicted 'logits' for some values of x (=Width)

$$\begin{aligned}
\text{logit}[\hat{\pi}(x = 22)] &= -1.412 \\
\text{logit}[\hat{\pi}(x = 24)] &= -0.417 \\
\text{logit}[\hat{\pi}(x = 26)] &= 0.577 \\
\text{logit}[\hat{\pi}(x = 28)] &= 1.571 \\
\text{logit}[\hat{\pi}(x = 30)] &= 2.566
\end{aligned}$$

plot logits as a function of W



1.3 The probit regression model

- GLM with binomial random component and probit link function
- nonlinear relationship between $\pi(x)$ and x:

$$\pi(x) = \Phi(\beta_0 + \beta_1 x)$$

- Φ is standard normal cumulative distribution function (cdf)
- using the 'probit' link function:

$$\Phi^{-1}[\pi(x)] = \text{probit}[\pi(x)] = \beta_0 + \beta_1 x$$

- β_1: represents the change in the 'probit' value (per unit change in x)
- other cdf's are possible: the logit transformation is simply the inverse function for the standard logistic cdf
- these are called *inverse CDF link functions*

R code

```
> fit <- glm(y ~ W, data=Crabs, family=binomial(link = "probit"))
> summary(fit)

Call:
glm(formula = y ~ W, family = binomial(link = "probit"), data = Crabs)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0519  -1.0494   0.5374   0.9126   1.6897

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.50196    1.50712  -4.978 6.44e-07 ***
W            0.30202    0.05804   5.204 1.95e-07 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 225.76  on 172  degrees of freedom
Residual deviance: 194.04  on 171  degrees of freedom
AIC: 198.04

Number of Fisher Scoring iterations: 5
```

results probit regression model

- the model predicts:

$$\text{probit}[\hat{\pi}(x)] = -7.50196 + 0.30202 \times \text{Width}$$

- in terms of the probability success:

$$\hat{\pi}(x) = \Phi(-7.50196 + 0.30202 \times \text{Width})$$

- the predicted probability success for some values of x (=Width)

$$\hat{\pi}(x = 22) = 0.196$$
$$\hat{\pi}(x = 24) = 0.400$$
$$\hat{\pi}(x = 26) = 0.637$$
$$\hat{\pi}(x = 28) = 0.830$$
$$\hat{\pi}(x = 30) = 0.940$$

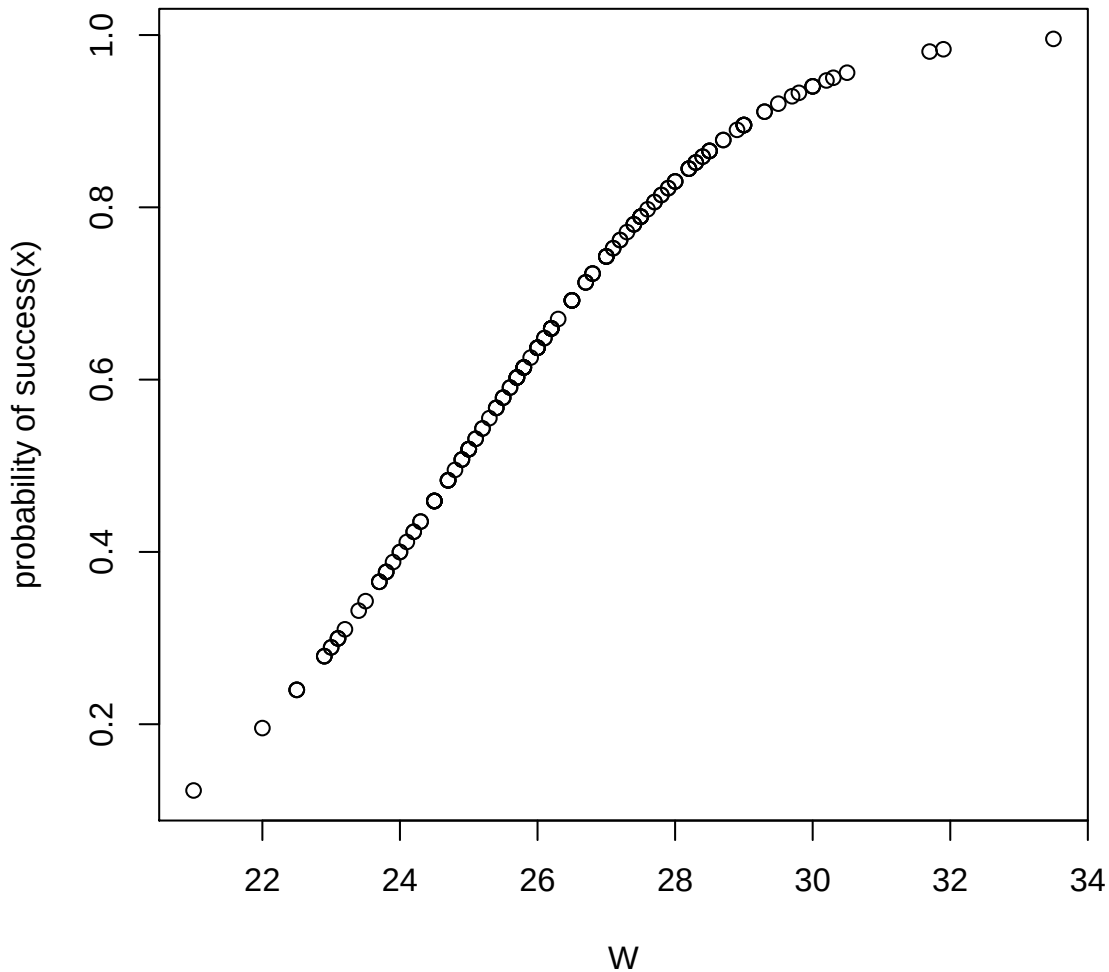
- computing predicted probabilities in R

```
> pnorm( -7.50196 + 0.30202*c(22,24,26,28,30) )

[1] 0.1955788 0.3999487 0.6370408 0.8301100 0.9404592
```

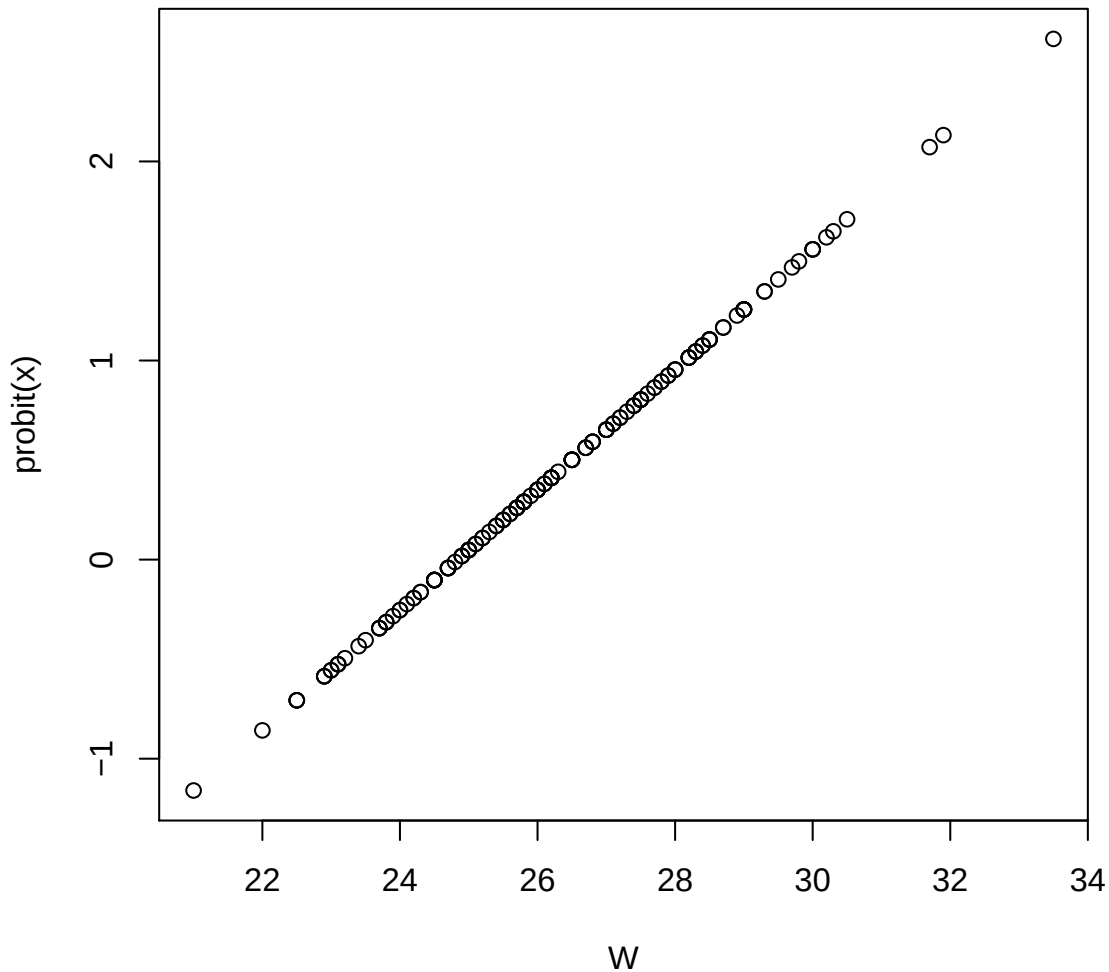
- or by using the predict() function with new data:

```
> # create `new' data in a data.frame
> W <- data.frame(W=c(22,24,26,28,30))
> W

   W
1 22
2 24
3 26
4 28
5 30


> # predict probabilities
> predict(fit, newdata=W, type="response")

        1         2         3         4         5
0.1955588 0.3999182 0.6370088 0.8300868 0.9404476
```

plot probability of success as a function of W



plot probits as a function of W



1.4 Regression with an ordinal response

cumulative probabilities

- ordinal variable y with K ordered levels: $k = 1, 2, 3, \ldots, K$
- probability $P(y = k)$: $\pi_1, \pi_2, \ldots, \pi_K$
- cumulative probabilities:

$$
\begin{aligned}
P(y \leq 1) &= \pi_1 \\
P(y \leq 2) &= \pi_1 + \pi_2 \\
P(y \leq 3) &= \pi_1 + \pi_2 + \pi_3 \\
&\ldots \\
P(y \leq K-1) &= \pi_1 + \pi_2 + \pi_3 + \ldots + \pi_{K-1} \\
P(y \leq K) &= 1
\end{aligned}
$$

cumulative logits

cumulative logits for $k = 1, 2, \ldots, K-1$:

$$\begin{aligned}\text{logit } P(y \leq k) &= \log\left(\frac{P(y \leq k)}{1 - P(y \leq k)}\right) \\ &= \log\left(\frac{P(y \leq k)}{P(y > k)}\right) \\ &= \log\left(\frac{\pi_1 + \ldots + \pi_k}{\pi_{k+1} + \ldots \pi_K}\right)\end{aligned}$$

- all K probabilities are used for each logit
- each logit resembles a binary logit with two categories:
  1. the categories 1 to k, and
  2. the categories $k+1$ to K.

proportional odds model

- a model that simultaneously uses all cumulative logits is

$$\log\left(\frac{\pi_1 + \ldots + \pi_k}{\pi_{k+1} + \ldots \pi_K}\right) = \beta_{0k} + \beta_1 x$$

  where $k = 1, 2, \ldots, K-1$.

- each cumulative logit has its own intercept β_{0k} ('threshold'); the $\{\beta_{0k}\}$ are increasing in k.

- the parameter β_1 has no subscript k: the effect of x is the same for each logit!

- it is similar to logistic regression for a binary response with outcomes $y \leq k$ (=1) and $y > k$ (=0).

- the response curves for $k = 1, 2, \ldots, K$ have the same shape, but are horizontally displaced from each other depending on the threshold (intercept) β_{0k}

parameterization 1

- the standard parameterization for the proportional odds model:

$$\log\left(\frac{\pi_1 + \ldots + \pi_k}{\pi_{k+1} + \ldots \pi_K}\right) = \beta_{0k} + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

  where $k = 1, 2, \ldots, K-1$.

- using this parameterization, a positive value for β_1 means that with increasing values for x, the odds increase of being less than a given value k:

  a positive coefficient implies increasing probability of being in lower-numbered categories (of the dependent variable y) with increasing values for x (holding everything else fixed).

- software: SAS (and the examples in Agresti's Book)

parameterization 2

- the alternative parameterization for the proportional odds model:

$$\log\left(\frac{\pi_1 + \ldots + \pi_k}{\pi_{k+1} + \ldots \pi_K}\right) = \beta_{0k} - (\beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p)$$

  where $k = 1, 2, \ldots, K-1$.

- using this parameterization, a positive value for β_1 means that with increasing values for x, the odds increase of being above a given value k:

  a positive coefficient implies increasing probability of being in higher-numbered categories (of the dependent variable y) with increasing values for x (holding everything else fixed).

- software: SPSS, R (`polr`) and lavaan

- this corresponds with a 'latent variable' interpretation of the ordinal dependent variable

example: mental impairment

- example from Agresti 2002: Table 7.5, 40 subjects, three variables:
  - y: mental impairment with four ordered levels: 1=well, 2=mild, 3=moderate, 4=impaired.
  - ses (1=high,0=low)
  - severity/number of important life events (e.g. birth of child, new job, divorce, . . .); treated as an interval variable.
- read in data in R:

```
> table.7.5 <-
+ read.table("http://www.da.ugent.be/datasets/Agresti2002.Table.7.5.dat",
+              header=TRUE)
```

- important: we need to 'declare' the mental variable as an 'ordered' variable; because there is no numeric code, we must also specify the ordering:

```
> table.7.5$mental <- ordered(table.7.5$mental,
+                             levels = c("well","mild","moderate","impaired"))
```

- 10 random cases:

```
> set.seed(1234)
> table.7.5[ sample(1:40, 10), ]
```

```
     mental ses life
5      well   0    2
25 moderate   0    0
24     mild   1    1
38 impaired   1    8
31 moderate   0    3
23     mild   1    3
1      well   1    1
8      well   1    3
22     mild   0    3
16     mild   0    1
```

- analysis using proportional odds model (logit scale):

```
> library(MASS)
> fit.polr <- polr(mental~ses +life, data=table.7.5)
> summary(fit.polr)
```

```
Call:
polr(formula = mental ~ ses + life, data = table.7.5)
```

```
Coefficients:
       Value Std. Error t value
ses  -1.1112     0.6109  -1.819
life  0.3189     0.1210   2.635

Intercepts:
                   Value   Std. Error t value
well|mild         -0.2819  0.6423    -0.4389
mild|moderate      1.2128  0.6607     1.8357
moderate|impaired  2.2094  0.7210     3.0644

Residual Deviance: 99.0979
AIC: 109.0979
```

- interpretation:
  - `life = 0.319`: for a 1-unit increase of life score, the estimated odds of mental impairment above any fixed level k are about $\exp(0.319) = 1.38$ times higher.
  - `ses = -1.111`: at the high ses level, the estimated odds of mental impairment above any fixed level k are about $\exp(-1.111) = 0.33$ times the estimated odds at the low ses level.

ordered probit regression

- cumulative probits:

$$\begin{aligned}\text{probit } P(y \leq k) &= \Phi^{-1}\left(P(y \leq k)\right) \\ &= \Phi^{-1}\left(\pi_1 + \ldots + \pi_k\right)\end{aligned}$$

- ordered probit regression model (using parameterization 2):

$$\Phi^{-1}\left(\pi_1 + \ldots + \pi_k\right) = \beta_{0k} - \left(\beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p\right)$$

  where $k = 1, 2, \ldots, K-1$.

- using this parameterization, a positive value for, say, β_1 means that with increasing values for x_1, the probability of being in a higher-numbered category, $P(y > k)$, increases

- this is similar as in ordinary regression, where a positive regression coefficient for x implies a positive relationship between y and x

example: mental impairment (probit)

- analysis using probit ordered regression (probit scale)

```
> fit.polr <- polr(mental~ses +life, data=table.7.5, method="probit")
> summary(fit.polr)

Call:
polr(formula = mental ~ ses + life, data = table.7.5, method = "probit")

Coefficients:
       Value Std. Error t value
ses  -0.6834    0.36411  -1.877
life  0.1954    0.06887   2.837

Intercepts:
                  Value   Std. Error t value
well|mild         -0.1612  0.3797    -0.4245
mild|moderate      0.7456  0.3849     1.9371
moderate|impaired  1.3392  0.4102     3.2648

Residual Deviance: 98.8397
AIC: 108.8397
```

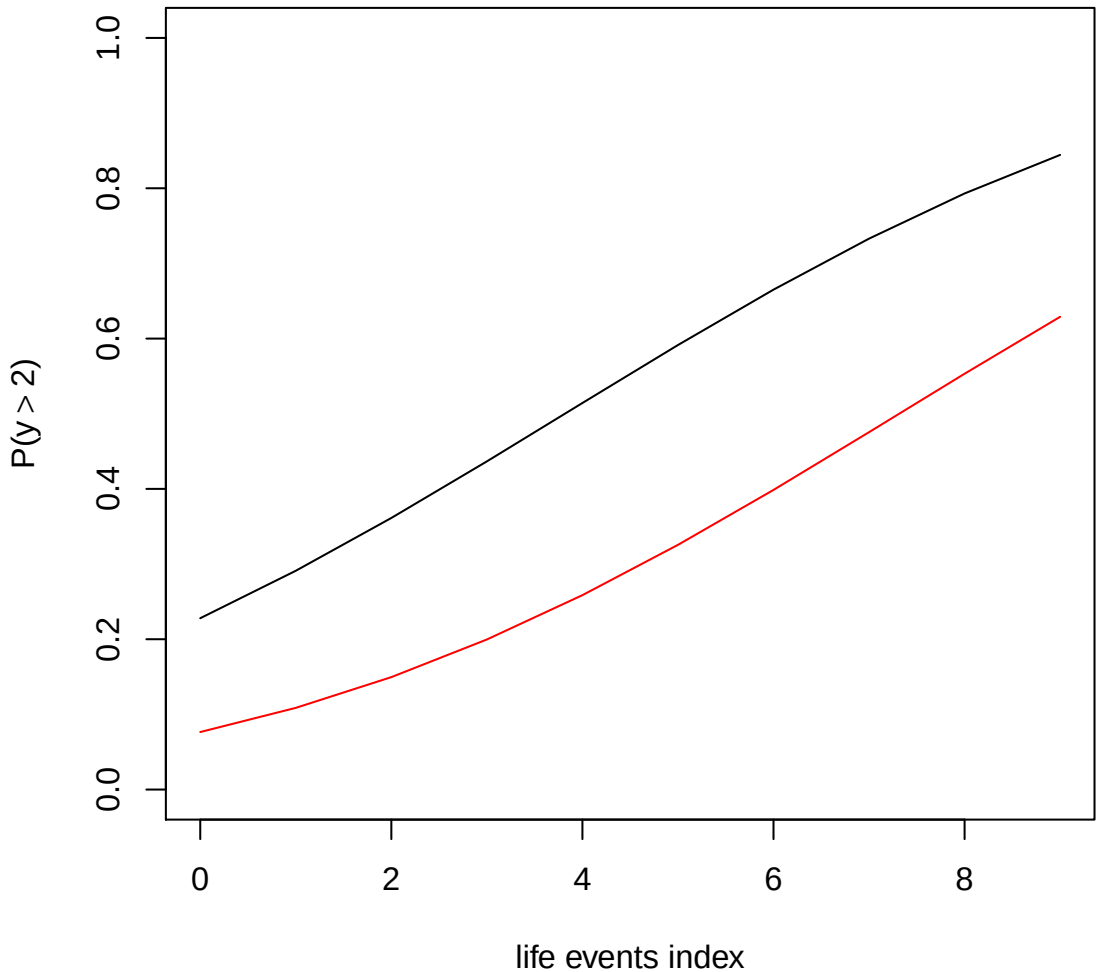
- interpretation:
  - `life = 0.1954`: for a 1-unit increase of life score, the estimated probit for $P(y > k)$ for mental impairment increases with 0.1954
  - `ses = -0.6834`: at the high ses level, the estimated cumulative probit for mental impairment is about -0.6834 lower than at the low ses level
  - the cumulative probit value is the z-score for $P(y > k)$
  - the odds can not be expressed as a simple function of the regression coefficients
- approximate relationship between logit and probit coefficients:

```
> coef(fit.polr) * 1.7

       ses       life
-1.1617096  0.3320972
```

plot of predicted (cumulative) probabilities

- make a plot, manually computing $P(y > k)$ for an arbitrary k:

```
> life.ses0 <- data.frame(ses=rep(0,10),life=c(0:9))
> prob.ses0 <- predict(fit.polr, newdata=life.ses0, type="probs")
> prob.ses0

          well      mild  moderate   impaired
1  0.43597454 0.3360801 0.1376880 0.09025734
2  0.36072016 0.3482159 0.1647149 0.12634899
3  0.29051331 0.3481649 0.1898765 0.17144525
4  0.22746025 0.3359325 0.2109178 0.22568944
5  0.17294576 0.3127853 0.2257672 0.28850177
6  0.12757286 0.2810357 0.2328708 0.35852071
7  0.09121819 0.2436619 0.2314602 0.43365969
8  0.06317660 0.2038508 0.2216893 0.51128332
9  0.04235452 0.1645584 0.2046066 0.58848055
10 0.02747037 0.1281718 0.1819698 0.66238806


> life.ses1 <- data.frame(ses=rep(1,10),life=c(0:9))
> prob.ses1 <- predict(fit.polr, newdata=life.ses1, type="probs")

> plot(0:9, rowSums(prob.ses0[,c("moderate","impaired")]),
+ ylim=c(0,1),xlim=c(0,9),type="l",
+ xlab="life events index", ylab=expression("P(y" > "2)"))
> lines(0:9, rowSums(prob.ses1[,c("moderate","impaired")]), col="red")
```


P(y > 2)
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
life events index

the underlying latent response variable approach

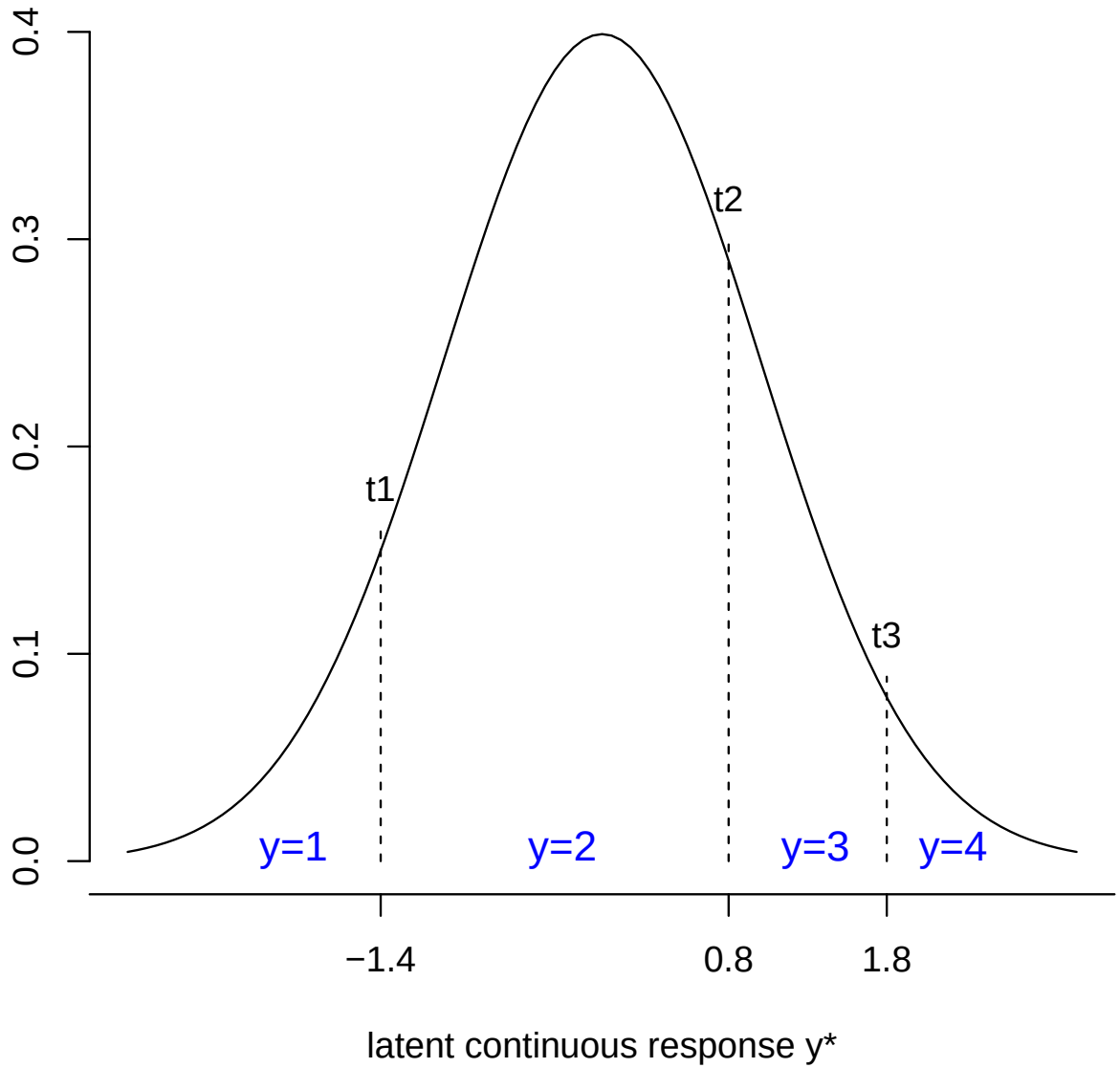
- an elegant way to think about ordinal variables is that they are a crude approximation of an underlying continuous variable
- since this continuous variable is not directly observed, we call it a latent response variable, denoted by $y^\star$ (y star)
- relationship between ordinal y (with K response categories) and $y^\star$:

$$y = k \quad \Longleftrightarrow \quad \tau_{k-1} < y^\star < \tau_k$$

  for the categories $k = 1, 2, \ldots, K-1$; furthermore, we let $\tau_0 = -\infty$ and $\tau_K = +\infty$
- the τ_k values are called cutpoints or *thresholds*
- typical assumption: $y^\star$ is normally distributed with mean zero, and unit variance:

$$y^\star \sim \mathrm{N}(0, 1)$$

example: ordinal variable with $K = 4$ response categories



the latent response variable regression model

- the latent response variable regression model:

$$\begin{aligned} y^\star &= \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \epsilon \\ &= \mathbf{X}\boldsymbol{\beta} + \epsilon \end{aligned}$$

- note that we observe $y = k$ when $y^\star$ falls between τ_{k-1} and τ_k; this implies that

$$\begin{aligned} P(y = k|\mathbf{X}) &= P(\tau_{k-1} < y^\star < \tau_k|\mathbf{X}) \\ &= P(\tau_{k-1} < \mathbf{X}\boldsymbol{\beta} + \epsilon < \tau_k|\mathbf{X}) \end{aligned}$$

- when we substract $\mathbf{X}\boldsymbol{\beta}$ within the inequality, we have

$$P(y = k|\mathbf{X}) = P(\tau_{k-1} - \mathbf{X}\boldsymbol{\beta} < \epsilon < \tau_k - \mathbf{X}\boldsymbol{\beta}|\mathbf{X})$$

- the probability that a random variable ϵ is between two values, is the difference between the CDF evaluated at these values; hence, we have

$$\begin{aligned} P(y = k|\mathbf{X}) &= P(\epsilon < \tau_k - \mathbf{X}\boldsymbol{\beta}|\mathbf{X}) - P(\epsilon < \tau_{k-1} - \mathbf{X}\boldsymbol{\beta}|\mathbf{X}) \\ &= \Phi(\tau_k - \mathbf{X}\boldsymbol{\beta}) - \Phi(\tau_{k-1} - \mathbf{X}\boldsymbol{\beta}) \end{aligned}$$

  where $\Phi(\tau_0 - \mathbf{X}\boldsymbol{\beta}) = 0$ and $\Phi(\tau_K - \mathbf{X}\boldsymbol{\beta}) = 1$

- the cumulative probabilities are defined as

$$P(y \leq k|\mathbf{X}) = \Phi(\tau_k - \mathbf{X}\boldsymbol{\beta}|\mathbf{X})$$

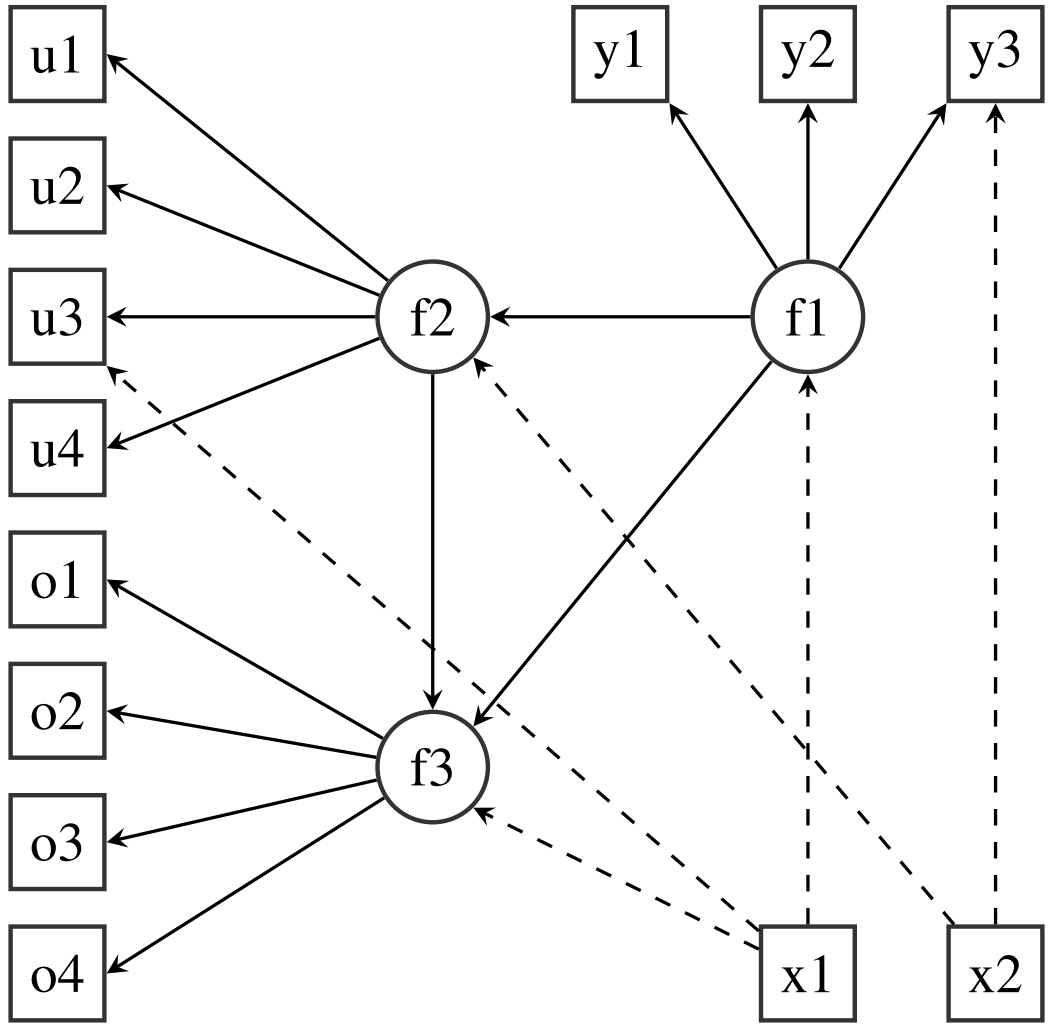
- in other words, this is all identical to the ordered probit regression model

- note that we implicitly assumed that $\beta_0 = 0$; if we enter β_0 in the model, we need to fix one of the thresholds to a constant, typically $\tau_1 = 0$

1.5 SEM with categorical (endogenous) variables: two approaches

- limited information approach
  - only univariate and bivariate information is used
  - estimation often proceeds in two or three stages; the first stages use maximum likelihood, the last stage uses (weighted) least squares
  - mainly developed in the SEM literature
  - perhaps the best known implementation is in Mplus
- full information approach
  - all information is used
  - most practical: marginal maximum likelihood estimation
  - requires numerical integration (number of dimensions = number of latent variables)
  - mainly developed in the IRT literature (and GLMM literature)
  - only recently incorporated in modern SEM software

example SEM framework: u = binary, o = ordered, y = numeric



a limited information approach: the WLSMV estimator

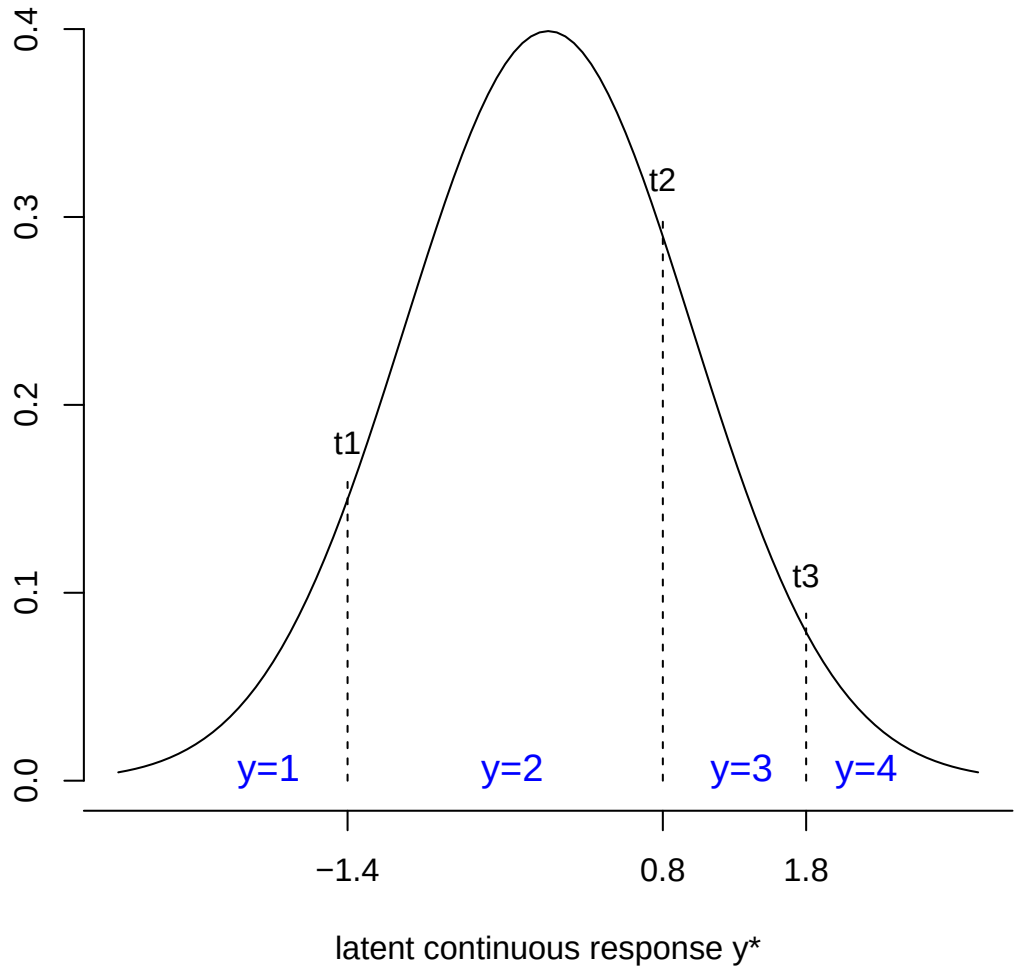
- developed by Bengt Muthén, in a series of papers; the seminal paper is

  > Muthén, B. (1984). A general structural equation model with dichotomous, ordered categorical, and continuous latent variable indicators. *Psychometrika, 49*, 115–132

- this approach has been the 'golden standard' in the SEM literature for almost three decades
- first available in LISCOMP (Linear Structural Equations using a Comprehensive Measurement Model), distributed by SSI, 1987 – 1997
- follow up program: Mplus (Version 1: 1998), currently version 7.11
- other authors (Jöreskog 1994; Lee, Poon, Bentler 1992) have proposed similar approaches (implemented in LISREL and EQS respectively)
- another great program: MECOSA (Arminger, G., Wittenberg, J., Schepers, A.) written in the GAUSS language (mid 90's)

stage 1 – estimating the thresholds (1)

- an observed variable y can often be viewed as a partial observation of a latent continuous response $y^\star$; eg ordinal variable with $K = 4$ response categories:



stage 1 – estimating the thresholds (2)

- estimating the thresholds: maximum likelihood using univariate data
- if no exogenous variables, this is just converting the cumulative proportions to z-scores

```
> # generate `ordered' data with 4 categories
> Y <- sample(1:4, size = 100, replace = TRUE)
> head(Y, 20)

 [1] 3 3 2 4 2 4 2 2 1 1 2 2 1 1 1 4 3 4 4 1


> prop <- table(Y)/sum(table(Y))
> prop

Y
   1    2    3    4
0.34 0.27 0.21 0.18


> cprop <- c(0, cumsum(prop))
> cprop

        1    2    3    4
0.00 0.34 0.61 0.82 1.00
```

```
> th <- qnorm(cprop)
> th
```

```
                  1          2          3          4
     -Inf -0.4124631  0.2793190  0.9153651        Inf
```

- in the presence of exogenous covariates, this is just ordered probit regression

```
> library(MASS)
> X1 <- rnorm(100); X2 <- rnorm(100); X3 <- rnorm(100)
> fit <- polr(ordered(Y) ~ X1 + X2 + X3, method = "probit")
> fit$zeta
```

```
       1|2        2|3        3|4
-0.4419159  0.2523076  0.8938680
```

stage 2 – estimating tetrachoric, polychoric, . . . , correlations

- estimate tetrachoric/polychoric/. . . correlation from bivariate data:
  - tetrachoric (binary – binary)
  - polychoric (ordered – ordered)
  - polyserial (ordered – numeric)
  - biserial (binary – numeric)
  - pearson (numeric – numeric)
- ML estimation is available (see eg. Olsson 1979 and 1982)
  - two-step: first estimate thresholds using univariate information only; then, keeping the thresholds fixed, estimate the correlation
  - one-step: estimate thresholds and correlation simultaneously
- if exogenous covariates are involved, the correlations are based on the residual values of $y^\star$ (eg bivariate probit regression)

stage 3 – estimating the SEM model

- third stage uses weighted least squares:

$$F_{WLS} = (\mathbf{s} - \hat{\boldsymbol{\sigma}})^\top \mathbf{W}^{-1} (\mathbf{s} - \hat{\boldsymbol{\sigma}})$$

  where $\mathbf{s}$ and $\hat{\boldsymbol{\sigma}}$ are vectors containing all relevant sample-based and model-based statistics respectively

- $\mathbf{s}$ contains: thresholds, correlations, optionally regression slopes of exogenous covariates, optionally variances and means of continuous variables

- the weight matrix $\mathbf{W}$ is (a consistent estimator of) the asymptotic (co)variance matrix of the sample statistics ($\mathbf{s}$)

- computing this weight matrix $\mathbf{W}$ is (sometimes) rather complicated

alternative estimators, standard errors, and test statistics

- in the weighted least squares framework, we can choose between three different choises for $\mathbf{W}$, leading to three different estimators:
  - estimator WLS: the full weight matrix $\mathbf{W}$ is used during estimation
  - estimator DWLS: only the diagonal of $\mathbf{W}$ is used during estimation
  - estimator ULS: $\mathbf{W}$ is replaced by the identity matrix ($\mathbf{I}$)
- two common types of standard errors:
  - 'classic' standard errors (based on the information matrix only)
  - 'robust' standard errors (using a sandwich type approach)
- four test statistics:
  - uncorrected, standard chi-square test statistic
  - mean adjusted test statistic (Satorra-Bentler type)
  - mean and variance adjusted test statistic (Satterthwaite type)
  - scaled and shifted test statistic (new in Mplus 6)

the Mplus legacy

- in Mplus, the 'default' estimator (for models with endogenous categorical variables) is termed WLSMV
- the term 'WLSMV' is widely used in the SEM literature
- in version 1 up to version 5 of Mplus, estimator WLSMV implies:
  - diagonally weighted least-squares estimation (DWLS)
  - robust standard errors
  - a mean and variance adjusted test statistic (hence, the MV extension)
- other available estimators (in Mplus) are
  - WLS (classical WLS, full weight matrix, classic standard errors and test statistic)
  - WLSM (DWLS + robust standard errors + mean-adjusted test statistic)
  - ULS, USLM and ULSMV (the latter two use the full weight matrix for computing standard errors and adjusted test statistics)

- since Mplus 6 (April 2010), the mean and variance adjusted test statistic was replaced by a ‘scaled and shifted’ test statistic
  - they still call this WLSMV
  - no need to adjust the degrees of freedom, so interpretation is easier
  - to get the ‘old’ behaviour, you need to set the ‘satterthwaite=on’ option

estimators, standard errors and test statistics in lavaan

- in lavaan, you can set your estimator, type of standard errors, and type of test statistic separately
- estimators (least squares framework):
  - `estimator="WLS"`
  - `estimator="DWLS"`
  - `estimator="ULS"`
- standard errors:
  - `se="standard"`
  - `se="robust"`
  - `se="bootstrap"`
- test statistics:
  - `test="standard"`

  - `test="Satorra.Bentler"`
  - `test="Satterthwaite"`
  - `test="scaled.shifted"`
  - `test="bootstrap` or `test="Bollen.Stine"`
- or you can use the Mplus style shortcuts
- `estimator="WLSMV"` implies
  - `estimator="DWLS"`
  - `se="standard"`
  - `test="scaled.shifted"` (following Mplus 6 and higher)
- `estimator="WLSMVS"` implies
  - `estimator="DWLS"`
  - `se="standard"`
  - `test="Satterthwaite"` (following older versions of Mplus)

- alternatives:
  - `estimator="WLSM"`
  - `estimator="ULSMV"`
  - `estimator="ULSM"`

using categorical variables in lavaan

- before you start, check the 'type' (or class) of the variables you will use in your model: are they numeric, or factor, or ordered, . . . ?
- in R, you can check the 'type' of a variable by typing

```
> x <- c(3,4,5)
> class(x)

[1] "numeric"


> x <- factor(x)
> class(x)

[1] "factor"


> x <- ordered(x)
> class(x)

[1] "ordered" "factor"
```

varTable

- a convenience function to screen the variables in lavaan is the 'varTable()' function:

```
> library(lavaan)
> varTable(HolzingerSwineford1939)
```

	name	idx	nobs	type	exo	user	mean	var	nlev	lna
1	id	1	301	numeric	0	0	176.555	11222.961	0	
2	sex	2	301	numeric	0	0	1.515	0.251	0	
3	ageyr	3	301	numeric	0	0	12.997	1.103	0	
4	agemo	4	301	numeric	0	0	5.375	11.915	0	
5	school	5	301	factor	0	0	NA	NA	2	Grant-White|Paste
6	grade	6	300	numeric	0	0	7.477	0.250	0	
7	x1	7	301	numeric	0	0	4.936	1.363	0	
8	x2	8	301	numeric	0	0	6.088	1.386	0	
9	x3	9	301	numeric	0	0	2.250	1.279	0	
10	x4	10	301	numeric	0	0	3.061	1.355	0	
11	x5	11	301	numeric	0	0	4.341	1.665	0	
12	x6	12	301	numeric	0	0	2.186	1.200	0	
13	x7	13	301	numeric	0	0	4.186	1.187	0	
14	x8	14	301	numeric	0	0	5.527	1.025	0	
15	x9	15	301	numeric	0	0	5.374	1.018	0	

using categorical variables in lavaan (2)

- two approaches to deal with 'ordered' (including binary) endogenous variables in lavaan:

  1. declare them as 'ordered' (using the `ordered()` function, which is part of base R) in your data.frame before you run the analysis;

     for example, if you need to declare four variables (say, item1, item2, item3, item3) as ordinal in your data.frame (called 'Data'), you can use something like:

     ```
     Data[,c("item1","item2","item3","item4")] <-
         lapply(Data[,c("item1","item2","item3","item4")], ordered)
     ```

  2. use the `ordered=` argument when using one of the fitting functions; for example, if you have four binary or ordinal variables (say, item1, item2, item3, item4), you can use:

     ```
     fit <- cfa(myModel, data=myData, ordered=c("item1","item2",
                                                "item3","item4"))
     ```

example: mental impairement

- here, the endogenous variable 'mental' has already been declared as ordered in the data frame (table.7.5)
- lavaan code:

```
> library(lavaan)
> model <- ' mental ~ ses + life '
> fit <- sem(model, data=table.7.5)
> summary(fit)

lavaan (0.5-17.700) converged normally after  17 iterations

  Number of observations                            40

  Estimator                                       DWLS      Robust
  Minimum Function Test Statistic                0.000       0.000
  Degrees of freedom                                 0           0
  Minimum Function Value               0.0000000000000
  Scaling correction factor                                     NA
  Shift parameter
    for simple second-order correction (Mplus variant)

Parameter estimates:
```

```
  Information                                 Expected
  Standard Errors                          Robust.sem

                   Estimate  Std.err  Z-value  P(>|z|)
Regressions:
  mental ~
    ses              -0.683    0.393   -1.739    0.082
    life              0.195    0.069    2.849    0.004

Intercepts:
    mental            0.000

Thresholds:
    mental|t1        -0.161    0.375   -0.429    0.668
    mental|t2         0.746    0.382    1.954    0.051
    mental|t3         1.339    0.424    3.162    0.002

Variances:
    mental            1.000
```

- compare this to the output of polr():

```
> fit.polr <- polr(mental~ses +life, data=table.7.5, method="probit")
> summary(fit.polr)

Call:
polr(formula = mental ~ ses + life, data = table.7.5, method = "probit")
```

```
Coefficients:
       Value Std. Error t value
ses  -0.6834    0.36411  -1.877
life  0.1954    0.06887   2.837

Intercepts:
                  Value   Std. Error t value
well|mild         -0.1612  0.3797    -0.4245
mild|moderate      0.7456  0.3849     1.9371
moderate|impaired  1.3392  0.4102     3.2648

Residual Deviance: 98.8397
AIC: 108.8397
```

- the estimates are very similar, despite the fact the polr() uses ML estimation, and lavaan uses DWLS
- the standard errors are slightly different; this is partly due to the estimation method (ML vs DWLS), but also because lavaan uses a so-called 'robust' method to compute the standard errors

specifying the thresholds in the model syntax

- if we need to impose restrictions on the thresholds, we need to specify them in the model syntax:
- thresholds are specified using the | (bar) operator, and have fixed names: `t1`, `t2`, `t3`, ...
- in the example below, we fix the values of the second and third threshold:

```
> model <- '
+     mental ~ ses + life
+     # thresholds
+     mental | t1 + 0*t2 + 1*t3
+ '
> fit <- sem(model, data=table.7.5)
> summary(fit)

lavaan (0.5-17.700) converged normally after   9 iterations

  Number of observations                            40

  Estimator                                       DWLS      Robust
  Minimum Function Test Statistic                4.460       3.941
  Degrees of freedom                                 2           2
```

```
  P-value (Chi-square)                           0.108       0.139
  Scaling correction factor                                  1.293
  Shift parameter                                            0.492
    for simple second-order correction (Mplus variant)

Parameter estimates:

  Information                                 Expected
  Standard Errors                          Robust.sem

                   Estimate  Std.err  Z-value  P(>|z|)
Regressions:
  mental ~
    ses              -0.683    0.393   -1.739    0.082
    life              0.195    0.069    2.849    0.004

Intercepts:
    mental            0.000

Thresholds:
    mental|t1        -0.161    0.375   -0.429    0.668
    mental|t2         0.000
    mental|t3         1.000

Variances:
    mental            1.000
```

example: binary CFA version of Holzinger & Swineford

```
> # binary version of Holzinger & Swineford
> HS9 <- HolzingerSwineford1939[,c("x1","x2","x3","x4","x5",
+                                  "x6","x7","x8","x9")]
> HSbinary <- as.data.frame( lapply(HS9, cut, 2, labels=FALSE) )
> head(HSbinary)

  x1 x2 x3 x4 x5 x6 x7 x8 x9
1  1  2  1  1  2  1  1  1  2
2  2  1  1  1  1  1  1  1  2
3  1  1  1  1  1  1  1  1  1
4  2  2  2  1  2  1  1  1  1
5  2  1  1  1  1  1  1  1  1
6  2  1  1  1  1  1  1  2  2
```

```
> # single factor model
> model <- ' visual  =~ x1 + x2 + x3
+            textual =~ x4 + x5 + x6
+            speed   =~ x7 + x8 + x9 '
> # binary CFA
> fit <- cfa(model, data=HSbinary, ordered=names(HSbinary))
> summary(fit, fit.measures=TRUE)

lavaan (0.5-17.700) converged normally after  35 iterations
```

```
  Number of observations                                301

  Estimator                                          DWLS      Robust
  Minimum Function Test Statistic                  30.918      38.546
  Degrees of freedom                                   24          24
  P-value (Chi-square)                              0.156       0.030
  Scaling correction factor                                     0.866
  Shift parameter                                               2.861
    for simple second-order correction (Mplus variant)

Model test baseline model:

  Minimum Function Test Statistic                 582.533     469.769
  Degrees of freedom                                   36          36
  P-value                                           0.000       0.000

User model versus baseline model:

  Comparative Fit Index (CFI)                       0.987       0.966
  Tucker-Lewis Index (TLI)                          0.981       0.950

Root Mean Square Error of Approximation:

  RMSEA                                             0.031       0.045
  90 Percent Confidence Interval          0.000     0.059       0.014  0.070
  P-value RMSEA <= 0.05                             0.847       0.596

Weighted Root Mean Square Residual:
```

```
  WRMR                                           0.829       0.829

Parameter estimates:

  Information                                 Expected
  Standard Errors                           Robust.sem

                   Estimate  Std.err  Z-value  P(>|z|)
Latent variables:
  visual =~
    x1                1.000
    x2                0.900    0.188    4.788    0.000
    x3                0.939    0.197    4.766    0.000
  textual =~
    x4                1.000
    x5                0.976    0.118    8.241    0.000
    x6                1.078    0.125    8.601    0.000
  speed =~
    x7                1.000
    x8                1.569    0.461    3.403    0.001
    x9                1.449    0.409    3.541    0.000

Covariances:
  visual ~~
    textual           0.303    0.061    4.981    0.000
    speed             0.132    0.049    2.700    0.007
  textual ~~
```

```
    speed             0.076    0.046    1.656    0.098

Intercepts:
    x1                0.000
    x2                0.000
    x3                0.000
    x4                0.000
    x5                0.000
    x6                0.000
    x7                0.000
    x8                0.000
    x9                0.000
    visual            0.000
    textual           0.000
    speed             0.000

Thresholds:
    x1|t1            -0.388    0.074   -5.223    0.000
    x2|t1            -0.054    0.072   -0.748    0.454
    x3|t1             0.318    0.074    4.309    0.000
    x4|t1             0.180    0.073    2.473    0.013
    x5|t1            -0.257    0.073   -3.506    0.000
    x6|t1             1.024    0.088   11.641    0.000
    x7|t1             0.231    0.073    3.162    0.002
    x8|t1             1.128    0.092   12.284    0.000
    x9|t1             0.626    0.078    8.047    0.000

Variances:
```

```
    x1                0.592
    x2                0.670
    x3                0.640
    x4                0.303
    x5                0.336
    x6                0.191
    x7                0.778
    x8                0.453
    x9                0.534
    visual            0.408    0.112
    textual           0.697    0.101
    speed             0.222    0.094
```

```
> inspect(fit, "sampstat")
```

```
$cov
   x1     x2     x3     x4     x5     x6     x7     x8     x9
x1  1.000
x2  0.284  1.000
x3  0.415  0.389  1.000
x4  0.364  0.328  0.232  1.000
x5  0.319  0.268  0.138  0.688  1.000
x6  0.422  0.322  0.206  0.720  0.761  1.000
x7 -0.048  0.061  0.041  0.200  0.023 -0.029  1.000
x8  0.159  0.105  0.439 -0.029 -0.059  0.183  0.464  1.000
x9  0.165  0.210  0.258  0.146  0.183  0.230  0.335  0.403  1.000
```

```
$mean
x1 x2 x3 x4 x5 x6 x7 x8 x9
 0  0  0  0  0  0  0  0  0

$th
 x1|t1  x2|t1  x3|t1  x4|t1  x5|t1  x6|t1  x7|t1  x8|t1  x9|t1
-0.388 -0.054  0.318  0.180 -0.257  1.024  0.231  1.128  0.626
```

parameter matrices

```
> inspect(fit)

$lambda
   visual textul speed
x1      0      0     0
x2      1      0     0
x3      2      0     0
x4      0      0     0
x5      0      3     0
x6      0      4     0
x7      0      0     0
x8      0      0     5
x9      0      0     6

$theta
   x1 x2 x3 x4 x5 x6 x7 x8 x9
x1  0
x2  0  0
x3  0  0  0
x4  0  0  0  0
x5  0  0  0  0  0
x6  0  0  0  0  0  0
x7  0  0  0  0  0  0  0
x8  0  0  0  0  0  0  0  0
x9  0  0  0  0  0  0  0  0  0
```

```
$psi
        visual textul speed
visual  16
textual 19     17
speed   20     21     18

$nu
   intrcp
x1      0
x2      0
x3      0
x4      0
x5      0
x6      0
x7      0
x8      0
x9      0

$alpha
        intrcp
visual       0
textual      0
speed        0

$tau
      thrshl
x1|t1      7
x2|t1      8
```

```
x3|t1      9
x4|t1     10
x5|t1     11
x6|t1     12
x7|t1     13
x8|t1     14
x9|t1     15
```

tables: univariate

```
> lavTables(fit, dim = 1)
```

	id	lhs	rhs	nobs	obs.freq	obs.prop	est.prop	X2
1	1	x1	1	301	105	0.349	0.349	0
2	1	x1	2	301	196	0.651	0.651	0
3	2	x2	1	301	144	0.478	0.478	0
4	2	x2	2	301	157	0.522	0.522	0
5	3	x3	1	301	188	0.625	0.625	0
6	3	x3	2	301	113	0.375	0.375	0
7	4	x4	1	301	172	0.571	0.571	0
8	4	x4	2	301	129	0.429	0.429	0
9	5	x5	1	301	120	0.399	0.399	0
10	5	x5	2	301	181	0.601	0.601	0
11	6	x6	1	301	255	0.847	0.847	0
12	6	x6	2	301	46	0.153	0.153	0
13	7	x7	1	301	178	0.591	0.591	0
14	7	x7	2	301	123	0.409	0.409	0
15	8	x8	1	301	262	0.870	0.870	0
16	8	x8	2	301	39	0.130	0.130	0
17	9	x9	1	301	221	0.734	0.734	0
18	9	x9	2	301	80	0.266	0.266	0

tables: bivariate (only first four)

```
> head( lavTables(fit, dim = 2), 16)
```

	id	lhs	rhs	nobs	row	col	obs.freq	obs.prop	est.prop	X2
1	1	x1	x2	301	1	1	63	0.209	0.222	0.228
2	1	x1	x2	301	2	1	81	0.269	0.256	0.198
3	1	x1	x2	301	1	2	42	0.140	0.127	0.400
4	1	x1	x2	301	2	2	115	0.382	0.395	0.128
5	2	x1	x3	301	1	1	83	0.276	0.271	0.022
6	2	x1	x3	301	2	1	105	0.349	0.353	0.017
7	2	x1	x3	301	1	2	22	0.073	0.078	0.078
8	2	x1	x3	301	2	2	91	0.302	0.298	0.020
9	3	x1	x4	301	1	1	76	0.252	0.243	0.101
10	3	x1	x4	301	2	1	96	0.319	0.328	0.075
11	3	x1	x4	301	1	2	29	0.096	0.105	0.233
12	3	x1	x4	301	2	2	100	0.332	0.323	0.076
13	4	x1	x5	301	1	1	56	0.186	0.183	0.020
14	4	x1	x5	301	2	1	64	0.213	0.216	0.017
15	4	x1	x5	301	1	2	49	0.163	0.166	0.022
16	4	x1	x5	301	2	2	132	0.439	0.435	0.009

1.6 Multiple group analysis with categorical data

- when comparing the means of latent variables (with categorical indicators), we again need to establish measurement invariance
- strong measurement invariance implies:
  - equal factor loadings across groups
  - equal thresholds across groups
- until now, we have always (implicitly) fixed the scale of the residual variances (of the categorical indicators) or the scale factors to unity (depending on the parameterization: 'delta' or 'theta')
- when we fix the thresholds across groups, we can relax this restriction, and freely estimate either the residual variances or the scale factors in the second group, third group, ...
- lavaan uses the 'delta' parameterization by default

example: binary CFA version of Holzinger & Swineford

```
> # binary version of Holzinger & Swineford
> HS9 <- HolzingerSwineford1939[,c("x1","x2","x3","x4","x5",
+                                  "x6","x7","x8","x9")]
> HSbinary <- as.data.frame( lapply(HS9, cut, 2, labels=FALSE) )
> HSbinary$school <- HolzingerSwineford1939$school
> head(HSbinary)

  x1 x2 x3 x4 x5 x6 x7 x8 x9  school
1  1  2  1  1  2  1  1  1  2 Pasteur
2  2  1  1  1  1  1  1  1  2 Pasteur
3  1  1  1  1  1  1  1  1  1 Pasteur
4  2  2  2  1  2  1  1  1  1 Pasteur
5  2  1  1  1  1  1  1  1  1 Pasteur
6  2  1  1  1  1  1  1  2  2 Pasteur



> # single factor model
> model <- ' visual  =~ x1 + x2 + x3
+            textual =~ x4 + x5 + x6
+            speed   =~ x7 + x8 + x9 '
> # binary CFA
> fit <- cfa(model, data=HSbinary, group="school", ordered=names(HSbinary),
+            group.equal=c("thresholds", "loadings"))
> summary(fit, fit.measures=TRUE)
```

```
lavaan (0.5-17.700) converged normally after 148 iterations

  Number of observations per group
  Pasteur                                          156
  Grant-White                                      145

  Estimator                                       DWLS      Robust
  Minimum Function Test Statistic                55.900      70.626
  Degrees of freedom                                 51          51
  P-value (Chi-square)                            0.296       0.036
  Scaling correction factor                                   0.885
  Shift parameter for each group:
    Pasteur                                                   3.881
    Grant-White                                               3.607
    for simple second-order correction (Mplus variant)

Chi-square for each group:

  Pasteur                                        37.317      46.030
  Grant-White                                    18.583      24.596

Model test baseline model:

  Minimum Function Test Statistic               602.275     472.615
  Degrees of freedom                                 72          72
  P-value                                         0.000       0.000

User model versus baseline model:
```

```
  Comparative Fit Index (CFI)                    0.991       0.951
  Tucker-Lewis Index (TLI)                       0.987       0.931

Root Mean Square Error of Approximation:

  RMSEA                                          0.025       0.051
  90 Percent Confidence Interval          0.000  0.060       0.014  0.078
  P-value RMSEA <= 0.05                          0.859       0.459

Weighted Root Mean Square Residual:

  WRMR                                           1.115       1.115

Parameter estimates:

  Information                                 Expected
  Standard Errors                           Robust.sem

Group 1 [Pasteur]:

                   Estimate  Std.err  Z-value  P(>|z|)
Latent variables:
  visual =~
    x1                1.000
    x2                0.678    0.227    2.988    0.003
    x3                1.088    0.301    3.608    0.000
  textual =~
```

```
    x4                1.000
    x5                1.031    0.191    5.399    0.000
    x6                1.269    0.217    5.838    0.000
  speed =~
    x7                1.000
    x8                1.279    0.576    2.219    0.026
    x9                1.365    0.568    2.403    0.016

Covariances:
  visual ~~
    textual           0.290    0.081    3.568    0.000
    speed             0.138    0.068    2.012    0.044
  textual ~~
    speed             0.148    0.074    1.986    0.047

Intercepts:
    x1                0.000
    x2                0.000
    x3                0.000
    x4                0.000
    x5                0.000
    x6                0.000
    x7                0.000
    x8                0.000
    x9                0.000
    visual            0.000
    textual           0.000
    speed             0.000
```

```
Thresholds:
    x1|t1            -0.262    0.097   -2.708    0.007
    x2|t1            -0.095    0.065   -1.477    0.140
    x3|t1             0.122    0.102    1.204    0.229
    x4|t1             0.422    0.095    4.428    0.000
    x5|t1            -0.028    0.102   -0.276    0.782
    x6|t1             1.419    0.149    9.533    0.000
    x7|t1             0.000    0.101    0.000    1.000
    x8|t1             1.076    0.125    8.617    0.000
    x9|t1             0.615    0.108    5.710    0.000

Variances:
    x1                0.590
    x2                0.811
    x3                0.515
    x4                0.371
    x5                0.331
    x6               -0.012
    x7                0.743
    x8                0.580
    x9                0.522
    visual            0.410    0.156
    textual           0.629    0.139
    speed             0.257    0.153

Scales y*:
    x1                1.000
```

```
    x2                1.000
    x3                1.000
    x4                1.000
    x5                1.000
    x6                1.000
    x7                1.000
    x8                1.000
    x9                1.000


Group 2 [Grant-White]:

                   Estimate  Std.err  Z-value  P(>|z|)
Latent variables:
  visual =~
    x1                1.000
    x2                0.678    0.227    2.988    0.003
    x3                1.088    0.301    3.608    0.000
  textual =~
    x4                1.000
    x5                1.031    0.191    5.399    0.000
    x6                1.269    0.217    5.838    0.000
  speed =~
    x7                1.000
    x8                1.279    0.576    2.219    0.026
    x9                1.365    0.568    2.403    0.016
```

```
Covariances:
  visual ~~
    textual           0.112    0.052    2.142    0.032
    speed             0.237    0.296    0.802    0.423
  textual ~~
    speed             0.193    0.339    0.569    0.569

Intercepts:
    x1                0.000
    x2                0.000
    x3                0.000
    x4                0.000
    x5                0.000
    x6                0.000
    x7                0.000
    x8                0.000
    x9                0.000
    visual           -0.074    0.084   -0.877    0.380
    textual           0.468    0.120    3.904    0.000
    speed            -2.187    3.045   -0.718    0.473

Thresholds:
    x1|t1            -0.262    0.097   -2.708    0.007
    x2|t1            -0.095    0.065   -1.477    0.140
    x3|t1             0.122    0.102    1.204    0.229
    x4|t1             0.422    0.095    4.428    0.000
    x5|t1            -0.028    0.102   -0.276    0.782
    x6|t1             1.419    0.149    9.533    0.000
```

```
    x7|t1             0.000    0.101    0.000    1.000
    x8|t1             1.076    0.125    8.617    0.000
    x9|t1             0.615    0.108    5.710    0.000

Variances:
    x1                0.102
    x2                0.036
    x3                0.050
    x4                0.135
    x5                0.410
    x6                0.438
    x7               14.028
    x8                1.599
    x9               21.680
    visual            0.062    0.044
    textual           0.512    0.269
    speed             5.512   12.805

Scales y*:
    x1                2.466    0.877    2.812    0.005
    x2                3.931    1.899    2.070    0.038
    x3                2.847    1.168    2.437    0.015
    x4                1.244    0.366    3.399    0.001
    x5                1.024    0.246    4.165    0.000
    x6                0.890    0.153    5.833    0.000
    x7                0.226    0.297    0.762    0.446
    x8                0.307    0.293    1.049    0.294
    x9                0.177    0.205    0.861    0.389
```

residual variances and scaling factors

- under the default parameterization used by lavaan (the so-called 'delta' parameterization), we do NOT estimate the residual variances of categorical endogenous variables
- they are a function of other model parameters; in particular, they are defined as:

$$\text{diag}(\boldsymbol{\Theta}) = \frac{1}{\boldsymbol{\Delta}^2} - \text{diag}(\hat{\boldsymbol{\Sigma}}^{\star})$$

  where

  - $\hat{\boldsymbol{\Sigma}}^{\star} = \boldsymbol{\Lambda}(\mathbf{I} - \mathbf{B})^{-1}\boldsymbol{\Psi}(\mathbf{I} - \mathbf{B})'^{-1}\boldsymbol{\Lambda}'$ (note: without $\boldsymbol{\Theta}$)
  - the (squared) diagonal elements of $\boldsymbol{\Delta}$ represent the (conditional) variances $V(y^{\star}|\mathbf{x})$ (but in this example, we do not have any exogenous variables $\mathbf{x}$)
  - we refer to these diagonal elements as *scaling factors*
- in a single group, the scaling factors are fixed to unity (i.e.: $\Delta = \mathbf{I}$)

- however, in a multiple group analysis, we can freely estimate these scaling factors for all but the first group
- an alternative parameterization is the so-called 'theta' parameterization; here, the residual variances are free parameters, and the scaling factors $\boldsymbol{\Delta}$ are obtained from

$$\frac{1}{\boldsymbol{\Delta}^2} = \text{diag}(\hat{\boldsymbol{\Sigma}}^{\star}) + \boldsymbol{\Theta}$$

- in a multiple group analysis, the 'theta' parameterization fixes the residual variances (the diagonal elements of $\boldsymbol{\Theta}$) to unity in the first group, but estimates them in the other groups

using the theta parameterization

```
> fit <- cfa(model, data=HSbinary, group="school", ordered=names(HSbinary),
+            group.equal=c("thresholds", "loadings"), parameterization="theta")
> summary(fit)

lavaan (0.5-17.700) converged normally after 214 iterations

  Number of observations per group
  Pasteur                                         156
  Grant-White                                     145

  Estimator                                       DWLS      Robust
  Minimum Function Test Statistic                55.912      70.635
  Degrees of freedom                                51          51
  P-value (Chi-square)                           0.296       0.036
  Scaling correction factor                                  0.886
  Shift parameter for each group:
    Pasteur                                                  3.884
    Grant-White                                              3.610
    for simple second-order correction (Mplus variant)

Chi-square for each group:

  Pasteur                                        37.272      45.975
  Grant-White                                    18.640      24.660

Parameter estimates:
```

```
  Information                                 Expected
  Standard Errors                          Robust.sem

Group 1 [Pasteur]:

                   Estimate  Std.err  Z-value  P(>|z|)
Latent variables:
  visual =~
    x1                1.000
    x2                0.579    0.271    2.132    0.033
    x3                1.163    0.584    1.991    0.047
  textual =~
    x4                1.000
    x5                1.091    0.584    1.867    0.062
    x6                6.856   65.940    0.104    0.917
  speed =~
    x7                1.000
    x8                1.442    0.995    1.449    0.147
    x9                1.630    1.083    1.505    0.132

Covariances:
  visual ~~
    textual           0.625    0.284    2.199    0.028
    speed             0.208    0.128    1.620    0.105
  textual ~~
    speed             0.284    0.188    1.515    0.130
```

```
Intercepts:
    x1                0.000
    x2                0.000
    x3                0.000
    x4                0.000
    x5                0.000
    x6                0.000
    x7                0.000
    x8                0.000
    x9                0.000
    visual            0.000
    textual           0.000
    speed             0.000

Thresholds:
    x1|t1            -0.340    0.133   -2.568    0.010
    x2|t1            -0.106    0.074   -1.440    0.150
    x3|t1             0.170    0.145    1.173    0.241
    x4|t1             0.699    0.220    3.175    0.002
    x5|t1            -0.051    0.177   -0.287    0.774
    x6|t1            12.810  121.070    0.106    0.916
    x7|t1            -0.000    0.117   -0.002    0.999
    x8|t1             1.411    0.321    4.400    0.000
    x9|t1             0.853    0.241    3.539    0.000

Variances:
    x1                1.000
    x2                1.000
```

```
    x3                1.000
    x4                1.000
    x5                1.000
    x6                1.000
    x7                1.000
    x8                1.000
    x9                1.000
    visual            0.695    0.450
    textual           1.716    1.032
    speed             0.346    0.278

Scales y*:
    x1                0.768
    x2                0.901
    x3                0.718
    x4                0.607
    x5                0.573
    x6                0.111
    x7                0.862
    x8                0.762
    x9                0.722



Group 2 [Grant-White]:

                  Estimate  Std.err  Z-value  P(>|z|)
Latent variables:
```

```
  visual =~
    x1                1.000
    x2                0.579    0.271    2.132    0.033
    x3                1.163    0.584    1.991    0.047
  textual =~
    x4                1.000
    x5                1.091    0.584    1.867    0.062
    x6                6.856   65.940    0.104    0.917
  speed =~
    x7                1.000
    x8                1.442    0.995    1.449    0.147
    x9                1.630    1.083    1.505    0.132

Covariances:
  visual ~~
    textual           0.243    0.136    1.793    0.073
    speed             0.368    0.484    0.760    0.447
  textual ~~
    speed             0.384    0.700    0.548    0.584

Intercepts:
    x1                0.000
    x2                0.000
    x3                0.000
    x4                0.000
    x5                0.000
    x6                0.000
    x7                0.000
```

```
    x8                0.000
    x9                0.000
    visual           -0.096    0.109   -0.881    0.378
    textual           0.777    0.277    2.807    0.005
    speed            -2.619    3.790   -0.691    0.490

Thresholds:
    x1|t1            -0.340    0.133   -2.568    0.010
    x2|t1            -0.106    0.074   -1.440    0.150
    x3|t1             0.170    0.145    1.173    0.241
    x4|t1             0.699    0.220    3.175    0.002
    x5|t1            -0.051    0.177   -0.287    0.774
    x6|t1            12.810  121.070    0.106    0.916
    x7|t1            -0.000    0.117   -0.002    0.999
    x8|t1             1.411    0.321    4.400    0.000
    x9|t1             0.853    0.241    3.539    0.000

Variances:
    x1                0.173    0.149
    x2                0.045    0.051
    x3                0.097    0.114
    x4                0.380    0.492
    x5                1.280    1.104
    x6               35.869  674.862
    x7               20.071   57.532
    x8                2.794    4.814
    x9               43.937  111.617
    visual            0.105    0.083
```

```
    textual            1.432    1.117
    speed              7.821   19.106

Scales y*:
    x1                 1.894
    x2                 3.537
    x3                 2.044
    x4                 0.743
    x5                 0.579
    x6                 0.098
    x7                 0.189
    x8                 0.229
    x9                 0.124
```

1.7 Full information approach: marginal maximum likelihood

- origins: IRT models (eg Bock & Lieberman, 1970) and GLMMs
- the *marginal* likelihood for the response vector $\mathbf{y}_i$ can be written as

$$f(\mathbf{y}_i|\mathbf{x}_i;\boldsymbol{\theta}) = \int_{D(\boldsymbol{\eta})} f(\mathbf{y}_i|\boldsymbol{\eta},\mathbf{x}_i;\boldsymbol{\theta}) f(\boldsymbol{\eta}|\mathbf{x}_i;\boldsymbol{\theta}) d\boldsymbol{\eta}$$

  where $\mathbf{y}_i$ are observed endogenous variables, $\mathbf{x}_i$ are observed exogenous covariates, and $\boldsymbol{\eta}$ are latent variables; $D(\boldsymbol{\eta})$ is the domain of integration; $\boldsymbol{\theta}$ is the parameter vector

- numerical integration
  - Gauss-Hermite quadrature
  - adaptive quadrature
  - Laplace approximation
  - Monte Carlo integration
- some clever 'dimension reduction' techniques exist for special cases

the connection with IRT

- the theoretical relationship between SEM and IRT has been well documented:

  Takane, Y., & De Leeuw, J. (1987). On the relationship between item response theory and factor analysis of discretized variables. Psychometrika, 52, 393-408.

  Kamata, A., & Bauer, D. J. (2008). A note on the relation between factor analytic and item response theory models. Structural Equation Modeling, 15, 136-153. Jöreskog, K. G., & Moustaki, I. (2001). Factor analysis of ordinal variables: A comparison of three approaches. Multivariate Behavioral Research, 36, 347-387.

- IRT tends to be used more if the focus is on the scale and the item characteristics

- SEM tends to be used more if the focus is on structural relations among either observed or latent variables; with or without exogenous covariates

- in lavaan (since 0.5-16): `estimator="MML"`

when are they equivalent?

- probit (normal-ogive) versus logit: both metrics are used in practice
- a single-factor CFA on binary items is equivalent to a 2-parameter IRT model (Birnbaum, 1968):
  - in CFA: λ_i, τ_i and θ_i are the factor loadings, the thresholds, and the residual variances)
  - in IRT: α_i and β_i are item discrimination and difficulty respectively
  - for a standardized factor: $\alpha_i = \lambda_i / \sqrt{\theta_i}$ and $\beta_i = \tau_i / \lambda_i$
- a single-factor CFA on polychotomous (ordinal) items is equivalent to the graded response model (Samejima, 1969)
- there is no CFA equivalent for the 3-parameter model (with a guessing parameter)
- the Rasch model is equivalent to a single-factor CFA on binary items, but where all factor loadings are constrained to be equal (and the probit metric is converted to a logit metric)

1.8 PML: pairwise maximum likelihood

- special case of the broader framework of 'composite' maximum likelihood
  - key idea: the complex likelihood is broken down as a (weighted) product of component likelihoods which are easier to handle (computationally)
  - composite ML estimators are asymptotically unbiased, consistent, and normally distributed
  - key references:

    Lindsay, B. (1998). Composite likelihood methods. *Contemporary Mathematics, 80*, 221–239
    Varin, C. (2008). On composite marginal likelihoods. *Advances in Statistical Analysis, 92(1)*, 1–28.

- introduced in the SEM literature by Jöreskog & Moustaki (2001), De Leon (2005), Liu (2007)
- computational complexity can be kept low regardless the number of observed and latent variables

PML: pairwise maximum likelihood

- in PML, the log-likelihood is a sum of $p^\star = p(p-1)/2$ components, each component being the bivariate log-likelihood of two observed variables:

$$pl(\boldsymbol{\theta}) = \sum_{k<l} \ln L(\boldsymbol{\theta}; (\mathbf{y}_k, \mathbf{y}_l))$$

- a recent simulation study illustrates the many pleasant properties of PML:
  - bias and MSE of PML estimators and their (sandwich type) standard errors are found to be small in all experimental conditions, and decreasing with the sample size
  - Katsikatsou, M., Moustaki, I., Yang-Wallentin, F., & Jöreskog, K. G. (2012). Pairwise likelihood estimation for factor analysis models with ordinal data. *Computational Statistics & Data Analysis, 56(12)*, 4243–4258.
- a follow-up study illustrates how PML can be used in a 'large' SEM setting (7 latent variables, many indicators)

available software for the PML approach

- commercial software:
  - none
- non-commercial, open-source software
  - R package **lavaan** (since 0.5-11, dec 2012)
  - `estimator="PML"`