

Next Presentation:

# A Better Way to Flip (Transpose) a SAS® Dataset

Columbus, Ohio  
Sept 22-24, 2013

Columbus, Ohio  
Sept 22-24, 2013

Columbus, Ohio  
Sept 22-24, 2013

Columbus, Ohio  
Sept 22-24, 2013

Columbus, Ohio  
Sept 22-24, 2013

Presenter: **Arthur Tabachneck**



**Art holds a PhD from Michigan State University, has been a SAS user since 1974, is president of the Toronto Area SAS Society and has received such recognitions as the SAS Customer Value Award (1999), SAS-L Hall of Fame (2011), SAS Circle of Excellence (2012) and, in 2012, was recognized as being the first SAS user to have been awarded more than 10,000 points on the SAS Discussion Forums**

Columbus, Ohio  
Sept 22-24, 2013

Columbus, Ohio  
Sept 22-24, 2013

Columbus, Ohio  
Sept 22-24, 2013

Columbus, Ohio  
Sept 22-24, 2013

Columbus, Ohio  
Sept 22-24, 2013

# A Better Way to Flip (Transpose) a SAS® Dataset

**Art297**

**Arthur Tabachneck**  
**Thornhill, ON Canada**

**KSharp**

**Xia Ke Shan**  
**Beijing, China**

**Robert Vergile**  
**Lexington, MA**

**Astounding**

**Joe Whitehurst**  
**Atlanta, GA**

# Presentation Overview

- What the %transpose macro is
- How the macro works
- The macro's benefits



# What the %transpose macro is

- It's a SAS macro
- Looks and feels almost exactly like PROC TRANSPOSE
- Doesn't have all of the capabilities of PROC TRANSPOSE as it was designed for just one purpose: **to convert tall files into wide files**
- Has virtually the same options and statements as PROC TRANSPOSE + a few more
- Is easier to use than PROC TRANSPOSE
- Runs significantly faster than PROC TRANSPOSE



# Have you ever had to flip a SAS dataset from being tall to being wide?

i.e., from:

idnum	date	var1
1	2001JAN	SD
1	2001FEB	EF
1	2001MAR	HK
2	2001JAN	GH
2	2001APR	MM
2	2001MAY	JH



# flipping a SAS dataset

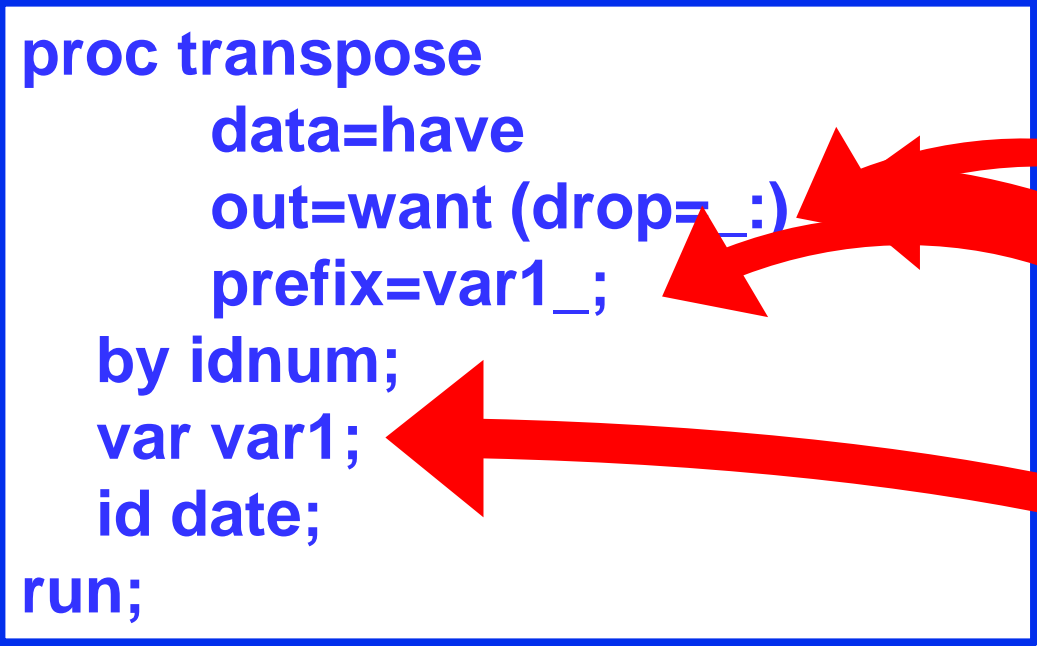
to:

idnum	var1_2001JAN	var1_2001FEB	var1_2001MAR	var1_2001APR	var1_2001MAY
1	SD	EF	HK		
2	GH			MM	JH



# if you have, you are probably already familiar with PROC TRANSPOSE

```
proc transpose  
  data=have  
  out=want (drop=_) ;  
  prefix=var1_ ;  
by idnum ;  
var var1 ;  
id date ;  
run ;
```



Not hard, but you need to know  
which are options  
which are statements

that you have to "drop" unwanted system variables  
and that you have to specify a prefix



would you be interested in knowing how to obtain the same result with the following code?

it may look like the PROC TRANSPOSE code, but:

**No system variables to drop**

```
%transpose( data=have,      out=want,  
            by=idnum,      var=var1,  
            id=date,       delimiter=_)
```

**No need for a prefix (var names automatically included)**

**No need to differentiate between options and statements as they are all of the form: parameter=value,**

**easier to code (less to type)**

**runs 10 times faster than PROC TRANSPOSE**





if you needed to flip a more complex  
SAS dataset from being tall to being wide  
i.e., from:

idnum	date	var1	var2
1	31MAR2013	1	SD
1	30JUN2013	2	EF
1	30SEP2013	3	HK
1	31DEC2013	4	HL
2	31MAR2013	5	GH
2	30JUN2013	6	MM
2	30SEP2013	7	JH
2	31DEC2013	8	MS



# flipping a more complex SAS dataset

to:

idnum	var1 Qtr1	var2 Qtr1	var1 Qtr2	var2 Qtr2	var1 Qtr3	var2 Qtr3	var1 Qtr4	var2 Qtr4
1	1	SD	2	EF	3	HK	4	HL
2	5	GH	6	MM	7	JH	8	MS



Again, you could use PROC TRANSPOSE  
but it would require two steps

First you have to make the table even taller (i.e.,  
one record for each *by variable* and *var* combination)

```
proc transpose data=have out=tall ;  
  by idnum date;  
  var var1-var2;  
  format date qtr1.;  
run;
```



That will create a taller file  
(i.e., 1 record for each *by variable* and *var combination*)

idnum	date	_NAME_	COL1
1	1	var1	1
1	1	var2	SD
1	2	var1	2
1	2	var2	EF
1	3	var1	3
1	3	var2	HK
1	4	var1	4
1	4	var2	HL
2	1	var1	5
2	1	var2	GH
2	2	var1	6
2	2	var2	MM
2	3	var1	7
2	3	var2	JH
2	4	var1	8
2	4	var2	MS



That will create a taller file  
(with var names now in **NAME** and values in **COL1**)

idnum	_NAME_	COL1
1	var1	1
1	var2	SD
1	var1	2
1	var2	3
1	var1	4
1	var2	5
1	var1	6
1	var2	7
1	var1	8
1	var2	MS
2	var1	9
2	var2	10
2	var1	11
2	var2	12
2	var1	13
2	var2	14
2	var1	15
2	var2	16
2	var1	17
2	var2	18
2	var1	19
2	var2	20



Then, to make the table wide  
(i.e., one record for each *by* variable)

You need to run PROC TRANSPOSE a 2<sup>nd</sup> time

```
proc transpose data=tall out=want (drop=_)  
  delimiter=_Qtr;  
  by idnum;  
  id _name_ date;  
  var col1;  
run;
```



# But there are some problems with the method

result:

idnum	var1 Qtr1	var2 Qtr1	var1 Qtr2	var2 Qtr2	var1 Qtr3	var2 Qtr3	var1 Qtr4	var2 Qtr4
1	1	SD	2	EF	3	HK	4	HL
2	5	GH	6	MM	7	JH	8	MS



The numeric variables are now character variables



and if the first idnum was missing data for one date

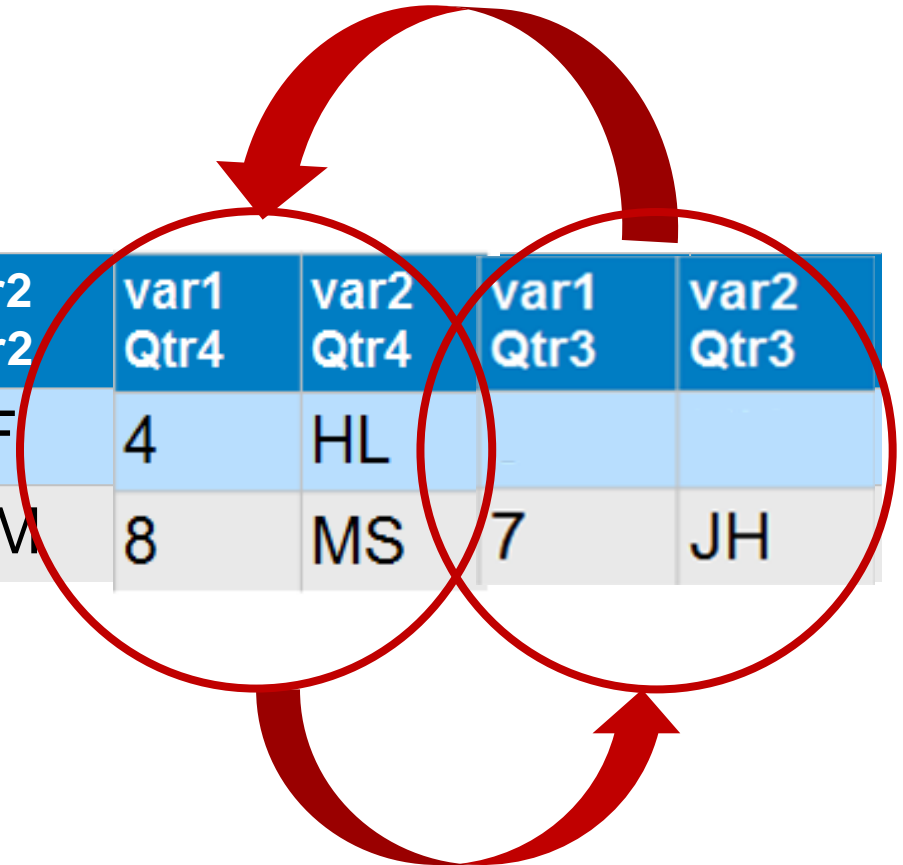
idnum	date	var1	var2
1	31MAR2013	1	SD
1	30JUN2013	2	EF
1	31DEC2013	4	HL
2	31MAR2013	5	GH
2	30JUN2013	6	MM
2	30SEP2013	7	JH
2	31DEC2013	8	MS





the output variable order will be a bit distorted

idnum	var1 Qtr1	var2 Qtr1	var1 Qtr2	var2 Qtr2	var1 Qtr4	var2 Qtr4	var1 Qtr3	var2 Qtr3
1	1	SD	2	EF	4	HL		
2	5	GH	6	MM	8	MS	7	JH



would you be interested in knowing how to obtain the right result with the following code?

```
%transpose(data=have, out=need, by=idnum,  
           id=date, format=qtr1., delimiter=_Qtr,  
           var=var1-var2)
```

How about if you knew that the macro:

only requires one step

only needs one pass through the data

doesn't produce distorted results

can run more than 50 times faster than  
**PROC TRANSPOSE**



# How the macro works if we have: dataset *have*

idnum	date	var1	var2
1	31MAR2013	1	SD
1	30JUN2013	2	EF
1	30SEP2013	3	HK
1	31DEC2013	4	HL
2	31MAR2013	5	GH
2	30JUN2013	6	MM
2	30SEP2013	7	JH
2	31DEC2013	8	MS



# How the macro works and we need: dataset *need*

idnum	var1 Qtr1	var2 Qtr1	var1 Qtr2	var2 Qtr2	var1 Qtr3	var2 Qtr3	var1 Qtr4	var2 Qtr4
1	1	SD	2	EF	3	HK	4	HL
2	5	GH	6	MM	7	JH	8	MS

and we submit:

- ```
%transpose(data=have, out=want, by=idnum,  
id=date, format=qtr1., delimiter=_Qtr,  
var=var1-var2)
```



## How the macro works:

- ①: if sort parameter had been equal to yes, the macro would have run PROC SORT and created a temporary sorted file
- ②: the macro drops unnecessary variables as early as possible
- ③: the macro creates and runs a datastep like the following:

```
data work.want (keep=last=1);
```

```
set work.have;
```

```
by idnum;
```

```
retain want_;
```

```
array have_;
```

```
array want_;
```

```
array want_num_;
```

```
if first.idnum then call missing(of want_chr(*));
```

```
__nchar=put(date,labelfmt.)*dim(have_chr);
```

```
do __i=1 to dim(have_chr);
```

```
  want_chr(__nchar+__i)=have_chr(__i);
```

```
end;
```

```
if first.idnum then call missing(of want_num(*));
```

```
__nnum=put(date,labelfmt.)*dim(have_num);
```

```
do __i=1 to dim(have_num);
```

```
  want_num(__nnum+__i)=have_num(__i);
```

```
end;
```

```
if last.idnum then output;
```

```
run;
```

labelfmt is a format, created by the macro, and reflects the ordered names of the transposed variables (i.e., from 1 to n)

```
Qtr4;  
Qtr4;
```

and you can use almost all the features  
that you can with PROC TRANSPOSE

plus some additional ones

```
%transpose(libname_in=, libname_out=,  
           data=,  
           by=,  
           var=,  
           id=,  
           format=,  
           copy=,  
           drop=,  
           out=,  
           prefix=,  
           autovars=,  
           var_first=,  
           delimiter=,  
           sort=,  
           guessingrows=)
```



# the %transpose macro's features

parameter: **libnames**

the names of the SAS libraries where your data reside and where you want the transposed file written

```
%transpose( libname_in=,      libname_out=,  
           data=,              out=,
```

\*\*\*\*\* FEATURE \*\*\*\*\*

While data and out can be assigned 1 or 2 level filenames

if your input or output files often use certain libraries you can assign them to these parameters as defaults

drop=;

guessingrows=;



# the %transpose macro's features

**parameter: autovars**

determines whether char(acter), num(eric) or all variables should be transposed if the var parameter is null

```
%transpose( libname_in=,      libname_out=,  
            data=,            out=,  
            by=,              prefix=,  
            var=,             autovars=,
```

\*\*\*\*\* FEATURE \*\*\*\*\*

Where PROC TRANSPOSE will only include all numeric variables if there is no var statement,

this parameter lets you indicate if you want all numeric variables, all character variables or simply all variables





# the %transpose macro's features

**parameter: var\_first**

**determines which is named first in transposed variables:**

**YES: prefix var name delimiter id value**

```
%transpose( libname_in=, libname_out=,  
            data=, out=,  
            by=, prefix=,  
            var=, autovars=,  
            id=, var_first=,  
            format=, delimiter=,  
            copy=, sort=,  
            drop=, guessingrows=)
```



# the %transpose macro's features

parameter: **var\_first**

determines which is named first in transposed variables:

YES: prefix var *name* delimiter id *value*

NO: prefix id *value* delimiter var *name*

```
%transpose( libname_in=, libname_out=,  
            data=, out=,  
            by=, prefix=,  
            var=, autovars=,  
            id=, var_first=,  
            format=, delimiter=,  
            copy=, sort=,  
            drop=, guessingrows=)
```



# the %transpose macro's features

parameter: **sort**

whether the input dataset should be sorted (YES or NO):

```
%transpose( libname_in=, libname_out=,  
            data=, out=,  
            by=, prefix=
```

\*\*\*\*\* FEATURE \*\*\*\*\*

If set to YES this parameter will insure that  
your data are automatically presorted

```
format=, delimiter=,  
copy=, sort=,  
drop=, guessingrows=)
```



# How many of you have ever:

- forgotten to run proc sort before running another proc that required sorted data?
- run proc sort before another proc only to discover that the file hadn't been sorted?
- run proc sort before another proc but didn't include a keep dataset option to limit the amount of data that had to be processed?
- run a proc that only used a few of a file's variables, but didn't include a keep dataset option to limit the amount of data that had to be processed?



Compare the performance of the following three sets of *almost* identical code run on a file with 40,000 records and 1,002 variables

```
PROC SORT data=have out=need;  
  by idnum date;  
run; took 6.96 seconds CPU time
```

```
DATA need;  
  set need;  
  _name_="var1";  
run; took 3.67 seconds CPU time
```

```
PROC TRANSPOSE data=need out=want (drop=_)  
  delimiter=_Qtr;  
  
  by idnum;  
  var var1;  
  id _name_ date;  
  format date Qtr1.;
```

run; took 1.51 seconds CPU time



Compare the performance of the following three sets of *almost* identical code run on a file with 40,000 records and 1,002 variables

```
PROC SORT data=have (keep=idnum date var1) out=need;  
  by idnum date;  
run; took 0.48 seconds CPU time
```

```
DATA need;  
  set need;  
  _name_="var1";  
run; took 0.06 seconds CPU time
```

```
PROC TRANSPOSE data=need out=want (drop=_)  
  delimiter=_Qtr;  
  
  by idnum;  
  var var1;  
  id _name_ date;  
  format date Qtr1.;;  
run; took 0.43 seconds CPU time
```



# Compare the performance of the following three sets of *almost* identical code run on a file with 40,000 records and 1,002 variables

```
%transpose( var=var1,  
            id=date,  
            format=Qtr1.,  
            sort=yes,  
            delimiter=_Qtr)
```

took 0.46 seconds CPU time  
i.e., twice as fast as the optimized code and  
25 times faster than the non-optimized code



# the %transpose macro's features

parameter: **guessingrows**

the number of rows to be read to determine the correct order for the set of transposed variables

\*\*\*\*\* FEATURE \*\*\*\*\*

With PROC TRANSPOSE the transposed variables will be in the order they are initially found in the data

this parameter controls the order based on the values found in the first *guessingrows*' records

**copy=,**  
**drop=,**

**sort=,**  
**guessingrows=)**





# the %transpose macro's features

**parameter: all parameters**

\*\*\*\*\* FEATURE \*\*\*\*\*

Since they are all macro *named parameters* you have direct control over their default values

If you set them to commonly used values, they don't have to be declared UNLESS you want to change their value

**var=,**

**id=,**

**format=,**

**copy=,**

**drop=,**

**datevar=,**

**var\_first=,**

**delimiter=,**

**sort=,**

**guessingrows=)**



# Benefits of the approach

- less typing thus fewer errors
- contains some features that would be nice to see available with all SAS procs
- easier to learn than PROC TRANSPOSE
- runs faster than PROC TRANSPOSE
- insures that you ALWAYS get the benefit of a critical SAS efficiency method
- more likely to provide the desired results



# Presentation Overview

- What the %transpose macro is ✓
- How the macro works ✓
- The macro's benefits ✓



**The macro, paper and Powerpoint can be found at:**

**<http://www.sascommunity.org>**

The macro, paper and Powerpoint can be found at:

<http://www.sascommunity.org>

The screenshot shows the sasCommunity.org website interface. On the left is a navigation sidebar with categories like 'Main Page', 'Blogs', 'Help', and 'Sasopedia'. The main content area features a 'Main Page' header, a welcome message, and a 'Popular Links' section. The 'Popular Links' section contains links for 'Proceedings', 'Tip of the Day', and 'More'. A red circle highlights the 'More' link, and a red arrow points from it to the 'More' link in the sidebar. A red oval highlights the URL at the top of the page.

Page Discussion Read View source View history Search

### Main Page

## Welcome to sasCommunity.org

A collaborative online community for SAS® users worldwide.

sasCommunity.org is an ongoing global community effort—created by SAS users for SAS users. This site is a place to find answers, share technical knowledge, collaborate on ideas and connect with others in the worldwide SAS Community. **Get started now** by [creating a UserID](#).

Follow us on [twitter](#)  
For daily SAS Tips, Web updates & more.

#### Popular Links

- Proceedings
- Tip of the Day
- More

...see also [\[read more\]](#) Submitted by Arthur Tabachneck. Contact me at my [Discussion Page](#).

# The macro, paper and Powerpoint can be found at:

## <http://www.sascommunity.org>

The screenshot shows the sasCommunity.org website interface. At the top right, there is a 'Log in / create account' link. Below it, there are navigation options: 'Page Discussion', 'Read', 'View source', 'View history', and a search box with a 'Search' button. The main content area is titled 'Popular Links' and contains the text 'Find What's Popular and New on sasCommunity.org:'. Below this, there are three columns of links:

- Get Knowledge, Tips and SAS Code**
  - Sasopedia
  - sasCommunity.org Tips of the Day
  - Presentations
  - SAS Global Forum 2013 Presentations
- Get Connected**
  - SAS Users on sasCommunity.org
  - Employment Opportunities
  - Events
  - User Groups
- Plus**
  - Connect with your peers
    - Originally designed for SAS Global Forum attendees, but anyone can participate
  - Try the new profile pages: Add your photo, send messages & more!
  - Discuss ideas for future SAS Global Forum events
  - View All New Features on sasCommunity.org

The link 'SAS Global Forum 2013 Presentations' is circled in red. On the left side, there is a sidebar with the sasCommunity.org logo and a navigation menu including: Main Page, Blogs, Help, Sasopedia, Sasopedia Detail, Language Elements, Procedures, Products, Topics, Popular Links, Pr, Tip, Mc, Co, Wt, Lo, Cr, Re, Upload file, Connect, Connect with Users, Contact Us, and Toolbox (What links here, Related changes, Special pages, Printable version, Permanent link).

At the bottom, there is a 'Get Help' section with links: Where to Start, Wiki Help, and Wiki Editing Basics. A 'Learn More' section includes: Welcome to sasCommunity.org, Web Site Features, and Resources for those who are new to SAS. A footer note states: 'If you did not find what you were looking for, or would like assistance in getting started on sasCommunity.org, please email us.'

The macro, paper and Powerpoint can be found at:

<http://www.sascommunity.org>

Log in / create account

Page **Discussion** Read View source View history  Search

## SAS Global Forum 2013 Presentations

If you are presenting at SAS Global Forum 2013, please [Create an Article for your SAS Global Forum Presentation](#).

Papers are now available online.

[Click here](#) to view 2009 Presentation Pages.  
[Click here](#) to view 2010 Presentation Pages.  
[Click here](#) to view 2011 Presentation Pages.  
[Click here](#) to view 2012 Presentation Pages.

- **A Better Way to Flip (Transpose) a SAS Dataset**
- Increase Your Productivity by Doing Less

Categories: [SAS Global Forum 2013](#) | Presentations

This page was last modified on 31 January 2013, at 17:13.  
This page has been accessed 1,009 times.

[Privacy policy](#) [About sasCommunity.org](#) [Terms of Use](#)

Powered By MediaWiki

**Questions?**



**Your comments and questions  
are valued and encouraged**

**Contact the Authors**

**Arthur Tabachneck, Ph.D.  
President, myQNA, Inc.  
Thornhill, ON  
art297@rogers.com**

**Joe Whitehurst  
High Impact Technologies  
Atlanta, GA  
joewhitehurst@gmail.com**

**Robert Virgile  
Robert Virgile Associates, Inc.  
Lexington, MA  
rvirgile@verizon.net**

**Xia Ke Shan  
Beijing, China  
xiakeshan@yahoo.com.cn**