STATA TECHNICAL BULLETIN

September 1991
STB-3

A publication to promote communication among Stata users

Subscriptions are available from Stata Corporation, email stata@stata.com, telephone 979-696-4600 or 800-STATAPC, fax 979-696-4601. Current subscription prices are posted at www.stata.com/bookstore/stb.html.

Previous Issues are available individually from StataCorp. See www.stata.com/bookstore/stbj.html for details.

Submissions to the STB, including submissions to the supporting files (programs, datasets, and help files), are on a nonexclusive, free-use basis. In particular, the author grants to StataCorp the nonexclusive right to copyright and distribute the material in accordance with the Copyright Statement below. The author also grants to StataCorp the right to freely use the ideas, including communication of the ideas to other parties, even if the material is never published in the STB. Submissions should be addressed to the Editor. Submission guidelines can be obtained from either the editor or StataCorp.

The *Stata Technical Bulletin* (ISSN 1097-8879) is published six times per year by Stata Corporation. Stata is a registered trademark of Stata Corporation.

Contents of this issue

		page
an1.1.	STB categories and insert codes (Reprint)	1
an9.	Change in associate editors	1
an10.	Stata available for DECstation	1
an11.	Stata X-Window driver available for SPARCstation	1
crc10.	Corrections and updates to roc and poisson commands	3
dm2.	Data conversion using DBMS/COPY and STAT/TRANSFER	3
dm2.1.	Vendors' response to review	7
gr6.	Lowess smoothing	7
gr7.	Using Stata graphs in the Windows 3.0 environment	9
os1.1.	Update on gphpen and color Postscript use	10
os3.	Using Intercooled Stata within DOS 5. 0	11
qs4.	Request for additional smoothers	12
sbe3.	Biomedical analysis with Stata: radioimmunoassay calculations	12
sed4.	Resistant normality check and outlier identification	15
sed5.	Enhancement of the Stata collapse command	18
sg1.1.	Correction to the nonlinear regression program	19
sg3.2.	Shapiro–Wilk and Shapiro–Francia tests	19
sg3.3.	Comment on tests of normality	20
sg3.4.	Review of tests of normality	20
sg3.5.	Comment on sg3.4 and an improved D'Agostino test	23
sg4.	Confidence intervals for t-test	25
snp2.	Friedman's ANOVA test & Kendall's coefficient of concordance	26
snp3.	Phi coefficient (fourfold correlation)	28
sqv1.2.	Additional logit regression diagnostic - Cook's Distance	28

an1.1 STB categories and insert codes (reprint)

Inserts in the STB are presently categorized as follows:

General Categories:

an	announcements	*ip*	instruction on programming
cc	communications & letters	*os*	operating system, hardware, & interprogram communication
dm	data management		
dt	data sets	*qs*	questions and suggestions
gr	graphics	*tt*	teaching
in	instruction	*zz*	not elsewhere classified

Statistical Categories:

sbe	biostatistics & epidemiology	*srd*	robust methods & statistical diagnostics
sed	exploratory data analysis	*ssa*	survival analysis
sg	general statistics	*ssi*	simulation & random numbers
smv	multivariate analysis	*sss*	social science & psychometrics
snp	nonparametric methods	*sts*	time-series, econometrics
sqc	quality control	*sxd*	experimental design
sqv	analysis of qualitative variables	*szz*	not elsewhere classified

In addition, we have granted one other prefix, *crc*, to the manufacturers of Stata for their exclusive use.

an9 Change in associate editors

Joseph Hilbe, Editor

It is with some sadness that I must announce Dr. Richard Goldstein has found it necessary to resign his position with the STB as Associate Editor. As many of you are aware, Dr. Goldstein is the Statistical Computing Software Review editor for the *American Statistician*, a publication of the American Statistical Association. His position requires a thoroughly objective perspective—both in fact and in appearance. He does not want others to believe that his association with the STB in any way conflicts with his ASA editorial obligations. As STB editor, I am in agreement that Dr. Goldstein must take this position. He still intends to submit inserts and provide advice when requested. His help has been invaluable to the genesis of the STB and I shall certainly accept his offer to provide continued input.

On the other hand, we are fortunate that Dr. Lawrence Hamilton has agreed to serve as Associate Editor. Lawrence Hamilton is Associate Professor of Sociology at the University of New Hampshire, where he teaches mainly statistics. He has written three Stata-oriented texts: *Statistics with Stata* (1990; second edition due 1992), *Modern Data Analysis* (1990), and *Regression with Graphics* (due late 1991), all with Brooks/Cole. His interests include exploratory, computer-intensive, and robust methods. I am pleased he is joining us.

an10 Stata available for DECstation

Ted Anderson, Marketing Director, CRC, 800-STATAPC

Stata is now available for the DEC Risc workstations running Ultrix. Included in the software is a DECwindows (X Windows) driver for displaying graphics either on the console or across the network. Prices for the DEC are the same as for all other Unix versions of Stata. Please contact CRC for more information. The product is shipping now.

an11 Stata X-Window driver available for SPARCstation

Ted Anderson, Marketing Director, CRC, 800-STATAPC

An X-Window (OpenWindows) driver is now available at no charge to owners of Stata 2.1 for the Sun SPARCstation. The driver provides all the same features available to SunView users and supports network terminals—so graphs may be displayed over the network when connected to a remote computer either through another SPARCstation or an X-terminal.

Updating your software is easy—call us and order the upgrade. There is a single 60K file that you copy into `/usr/local/stata/sun-4`. The software really is free if you will accept it on 3.5-inch diskette, in either Unix format or DOS format (which can be copied via NFS to the appropriate directory). If you insist on having it on cartridge tape, there is a $20 media charge.

crc10	Corrections and updates to roc and poisson commands

In *sbe1* published in STB-1, we introduced a new `poisson` command called `poisson2` that allowed specifying an offset (thus allowing analysis of rates). We have now had sufficient experience with `poisson2` to promote it to an official CRC offering. `poisson2` is now officially renamed `poisson`. We have also introduced a further modification: The reference model used for calculation of the chi-square now includes the offset as well as the constant. Thus, the chi-square test is now a test of

$$\mathrm{E}(y) = e^{a+xb+o}$$

versus

$$\mathrm{E}(y) = e^{a+o}$$

where o is the offset. This leads to a valid chi-square test whereas the previous version did not. We express our thanks to German Rodriguez of Princeton University for this tip.

We have made two corrections to `roc`, which is also used by `logiodds`. In the case of grouped data `roc` could produce an incorrect ROC curve. We also fixed a problem with weighted data. We express out thanks to Josie Pearson of the Clinical Research Center in England for spotting these problems.

dm2	Data format conversion using DBMS/COPY and STAT/TRANSFER

Joseph Hilbe, Editor, STB, FAX 602-860-1446

Stata users sometimes find it necessary to convert Stata data files into files formatted to be used by other programs; e.g., Paradox, dBASE, Lotus 123, Quattro, Excel, or even other statistical packages. Of course, the same is true in reverse—the need to convert files created by other programs into Stata `.dta` files. There are two major commercial data conversion packages currently on the market which explicitly address the Stata file format. The most comprehensive program is DBMS/COPY and its enhanced version DBMS/COPY PLUS. Both are published by Conceptual Software, Inc. (CSI) in Houston, Texas. The other program is STAT/TRANSFER by Circle Systems in Seattle, Washington. These packages will be reviewed with an emphasis on how each relates to Stata. A comparative summary will follow.

DBMS/COPY & DBMS/COPY PLUS – Version 2

CSI produces a relational database and integrated statistical package called PRODAS. Several years ago, responding to user requests, CSI began to develop a PC file transfer utility that would enable PRODAS users to convert files from other formats. It was named DBMS/COPY after "Database Management System." CSI has stated a long-range goal envisaging that DBMS/COPY will some day allow users to transfer data between every popular database, spreadsheet, and statistical package. With release 2 (October 1989), DBMS/COPY now converts between some 65 program formats, including ASCII.

DBMS/COPY only allows transfer between complete data sets; i.e., the entire data set is converted to a data set of another format. DBMS/COPY PLUS (Version 2, Feb. 1990) enables variable and observation selection as well as providing the user with a plethora of additional customization capabilities. For instance, when transferring a Stata file to a Paradox format, you can easily convert variable `x` in the `.dta` file to a factorial in the target file. Simply type the following in the DBMS/COPY PLUS Edit Window:

```
compute;
in=statafl.stata21 out=paradfl.db;
fac_var= exp(lgamma(x+1));
run;
```

The new Paradox file, `paradfl.db`, will now have a new variable named `fac_var` that is the factorial of variable `x` in the original Stata file.

The variety of functions allowed in the PLUS version appear nearly exhaustive for the majority of uses. There are 12 date, 6 financial, and 66 numeric functions including 26 mathematic, 21 probability, and 19 trigonometric functions. In addition, the user may modify, keep, delete, format, assign, label, or rename variables and may freely determine criteria for data selection.

DBMS/COPY PLUS allows the user to employ window menu selection or to write the code from scratch as above. Both packages have an extensive context sensitive help system which make it rather difficult for the astute user to err.

Data conversion is not without a downside. When using DBMS/COPY, I converted a Stata data set consisting of 10,000 observations and 35 variables to Paradox 3.5, dBASE III+, Lotus 123 2.2, SPSS/PC 4.0, and SAS PC files. The original Stata file

was 1,872,144 bytes while the converted files were respectively:

Paradox 3.5	2,560,581 bytes
dBASE III+	2,831,153 bytes
123 2.2	5,010,485 bytes
SPSS/PC	3,521,856 bytes
SAS PC	1,953,926 bytes

I then reconverted the files back to Stata format with the following results:

	bytes	ram	width
Original Stata file	1,872,144	1,826K	187
from Paradox 3.5	2,022,144	1,972K	202
from dBASE III+	2,062,144	2,011K	206
from 123 2.2	3,482,492	3,398K	348
from SPSS/PC 4.0	3,521,856	2,167K	222
from SAS PC	1,953,926	2,011K	206

Clearly some variables had been given greater width during conversion. SPSS/PC conversion automatically transfers SPSS system `_casenum`, `_date`, and `_weight_` variables which can be dropped (as was done prior to providing the SPSS statistics). Others simply widen float and string lengths. CSI techs claim that this is necessary to preserve precision. However, at first glance the seeming arbitrary inflation of converted data sets is rather disconcerting. But with some perspicacity, the user can reformat variables in the newly created data sets. Paradox and dBASE have distinct menus for file restructuring whereas spreadsheets such as 123 and Excel format by means of column width—which may easily, if not tediously, be altered. From within Stata you can use the compress command to reduce the width of each variable separately or to compress the entire data set with one word. Type

```
. compress varname
```

A by-variable compression listing is displayed on the screen. Variables may also be reformatted directly by using the `recast` command. For example,

```
. recast birthyr int
```

changes the variable `birthyr` into an integer, if consistent with the data in memory. Each of the above reconverted Stata data sets were compressed. The respective compressions resulted in data sets with the same values as the original Stata data set. Hence there were in fact no "real" alterations of data values when converting between various data formats.

DBMS/COPY has another feature which aids in Stata data transfer. The first time DBMS/COPY creates a Stata file, it also makes a second file with the same prefix name, but which contains only the variable types and names in the converted file. This file can be identified by its `.var` extension. You are then given the option of editing this `.var` ASCII file to change variable formats. Hence, if a spreadsheet allows only 8 byte floats as numbers, and a variable in your file is an integer, you may edit the `.var` file to format the variable as an `int`. Then simply run the conversion again. DBMS/COPY will look for and adapt the conversion to the edited `.var` file.

ASCII files can be converted to Stata files as well as the reverse. However, in order for the former to occur, the user must create a data dictionary containing parameter definitions and a description of each variable. Free and fixed format files are allowed and a menu system is available for assistance in the process of ASCII conversion.

There are four questions commonly asked on the Stata help line regarding data conversion:

1. What happens when the source file has duplicate variable names?
2. What if a source file variable is an illegal Stata variable name?
3. What if a source file variable name has a leading space?
4. What if the source file variable name is longer than 8 characters.

DBMS/COPY handles each in the following manner:

1. If the source file contains a duplicate variable name, DBMS/COPY provides the second instance (and third, etc.) with a new variable name; e.g., the second variable `x2` is renamed `x2_1`. A note is placed on the screen during conversion regarding the new variable name.
2. I tested two varieties of illegal Stata variable names. One variable name in the source file contained an illegal character in the middle of the name (`x&3`) while the other was throughly illegal (`%%&&`). DBMS/COPY immediately recognized the illegal characters and changed them to underscores. Hence, `x&3` is converted to `x_3` and `%%&&` to `____`. Since an series of underscores makes an odd, but not illegal, Stata name, you will probably want to `rename` the variable in Stata.

3. DBMS/COPY simply deletes the leading space in the source file variable name and converts the name as character-only.
4. DBMS/COPY keeps only the first eight characters of a variable name during Stata conversion. Longer variable names are chopped. If this results in a duplicate name, the method 1 for handling duplicate names is used to resolve the difficulty. For example, I named one variable in the source file `x12345678` and another `x12345679`. DBMS/COPY converted them respectively, after letting me know, to `x123456` and `x1234561`.

In short, DBMS/COPY does an excellent job at recognizing and dealing with Stata-strange variable names.

A special caveat should be given to Stata users when working with either of the CSI products. The version 2.0 disks were constructed to convert Stata 2.0 files. If you have Stata 2.1 or beyond, there will be instances when the Stata file you are attempting to convert will fail. For example, if your Stata file has variables formatted as `byte` and you are attempting to convert to another format, then a message will be presented on the DBMS/COPY (PLUS) screen that the Stata file is invalid. CSI has a 2.1 fix for this and will mail a copy if requested—but you must ask.

STAT/TRANSFER Version 1.4B

Stat/Transfer was initially designed to convert or transfer statistical data from mainframe SPSS and SAS files to the PC environment. Moreover, the creators of Stat/Transfer argue that the majority of PC database and spreadsheet programs have utilities that can read and write either Lotus "wk1" or dBASE "dbf" files. Hence, by supporting these two formats, the user has in effect access to conversion between virtually all PC database and spreadsheet system files.

Stat/Transfer provides the user with Kermit to allow transport of files between computers, including the downloading of mainframe SPSS data files. Saving as an SPSS Export file allows immediate conversion to, for example, a Stata format file—with the retention of variable and value labels. SAS mainframe files must first be converted to SPSS Export files by means of the SPSS utility TOSPSS.

Fortunately for the Stata user, Stat/Transfer adopts Stata 2.x as one of four statistical formats built into its program. The others are SPSS Export, Gauss, and Systat. Stat/Transfer also converts to and from Lotus 123 "wk1" and dBASE II, III, III+, and IV files.

Both DBMS/COPY and Stat/Transfer are menu-based systems. However, the latter also allows variable selection like the PLUS version of DBMS/COPY. It also appears to inflate Stata files when converted to another format, but when reconverted and compressed, are the same size as the original. For example, I converted the same Stata file as used when evaluating DBMS/COPY. Results for transfer to dBASE and 123 formats are as follows:

Original Stata file	1,872,144 bytes
dBASE format	2,601,154 bytes
123 format	5,261,831 bytes

When reconverted they reduced to

	bytes	ram	width
Original Stata file	1,872,144	1,826K	187
from dBASE III+	2,022,145	2,011K	206
from 123	1,961,965	1,914K	196

Both reconverted files compressed to the same bytes, ram, and width as the original Stata file upon compression.

How does Stat/Transfer deal with the four questions raised in the discussion of DBMS/COPY?

1. When converting duplicate source file variable names, Stat/Transfer simply transfers the duplicate name as found in the source file. The resultant Stata file thus has duplicate variable names. (The Stata user can fix things afterwards by renaming the first occurrence of the duplicate to a new, nonduplicated name. For example, if `myvar` appears more than once in the converted data set, typing "`rename myvar myvar1`" will change the name of the first `myvar` to `myvar1`. If `myvar` had appeared more than twice, then the trick can be repeated: "`rename myvar myvar2`", renaming what is now the first occurrence of `myvar` to `myvar2`.)
2. Illegal Stata variables are transferred to Stata as defined in the source file. Stata will produce an error message when the user attempts to directly address the illegal variable name. The problem is not easily fixed after conversion. One cannot generate a new variable based on the illegal name or use it directly in a statistical procedure; however you may indirectly refer to it. For example, if your data set contained the variables, in order, `x1`, `x&2`, and `x3`, you can refer to `x1-x3`. There is no way to refer to only `x&2`.
3. Stat/Transfer handles variable names that begin with a space in a manner similar to DBMS/COPY; it ignores the leading space.

4. Stat/Transfer also chops variable names in excess of eight characters, just as DBMS/COPY. However, if the first eight characters are the same, the resultant conversion will yield duplicate names and the user is back to the above mentioned problem.

When using Stat/Transfer, it is best to check and alter source file variable names prior to actual conversion.

I discovered a bug in Stat/Transfer 1.4 when converting between Stata and Lotus 123 files. If a Stata file contains a float variable where the first observation is missing, the converted 123 file will not acknowledge the missing value. Instead of a "." for the first observation for that variable, there is a blank space. When converting back to Stata format, the variable is not allowed as a selection option by Stat/Transfer. Hence it cannot be converted. This problem does not occur for string variables or for integers; nor does it occur at all when converting between Stata and dBASE. Circle Systems has since provided a fix for this problem with version 1.4B (7/10/91).

Summary

I believe that both DBMS/COPY and Stat/Transfer are excellent for accomplishing the tasks for which they were developed. DBMS/COPY allows for a wide range of data set conversions while the PLUS enhancement adds variable selection and the ability to create or modify variables with a host of functions. Stat/Transfer provides SPSS and SAS mainframe downloading capability together with effective conversion between programs which address 123 and dBASE files. It also allows menu driven variable selection. Both programs are well documented and have accessible phone support. However, I should note that although both utilities transfer records very quickly, Stat/Transfer 1.4B is slightly faster for some conversions. It took 39 seconds for both DBMS/COPY and DBMS/COPY PLUS to convert the test Stata data set of 10,000 records with 35 variables to dBASE III format, whereas it took Stat/Transfer 1.4B 37 seconds. Stat/Transfer 1.4 was much slower; it took 92 seconds to convert the same file. The test was performed on a 33 MHz 80486 computer with 16 megabytes of ram.

What are the criteria to determine which transfer utility, if any, is the best for your purposes? Perhaps the following can help:

1. If you use Stata for most of your statistical analyses and seldom, if ever, need to transfer data between Stata and other formats, you probably don't need a transfer utility. However, DBMS/COPY PLUS may prove useful if you wish to use various esoteric mathematical transformations on your Stata data set; e.g., converting one Stata file into another enhanced version.
2. If you do need to transfer data sets to other formats, whether between database/spreadsheets and Stata or between Stata and other statistical packages, then a conversion utility will most likely save you both time and money.
3. If you download files from mainframe SPSS or SAS, Stat/Transfer appears to be an ideal program.
4. If you use 123 or dBASE compatible programs and use Stata as your foremost statistical package, Stat/Transfer should prove adequate.
5. If you work strictly within the PC domain and use a variety of statistical, spreadsheet, and database packages, then DBMS/COPY may be more valuable. Of course, the PLUS version adds so many features that I suggest you use it.

(6) If cost is a factor, consider the following retail prices:

DBMS/COPY	$195.00 (see Addendum)
DBMS/COPY PLUS	$295.00
STAT/TRANSFER	$ 90.00 (with academic discount, $50.00)

Addendum

Since this article was written, I have learned that Conceptual Software has agreed to sell its marketing and technical support interests in DBMS/COPY PLUS to SPSS, Inc. DBMS/COPY, i.e., the non-PLUS version, will be discontinued. Conceptual Software will still continue to write the software enhancements; SPSS will provide all marketing, sales and technical support. SPSS management has told me that it intends to expand support for many other packages and that it hopes DBMS/COPY PLUS will find its way into a larger domestic as well as international markets.

You may order or request information on the software by contacting

DBMS/COPY PLUS

SPSS Inc.
444 N. Michigan Ave.
Chicago, IL 60611
(800) 543-2185

STAT/TRANSFER

Computing Resource Center
1640 Fifth Street
Santa Monica, CA 90401
(800) 782-8272
(213) 393-7551 (Fax)

dm2.1 Vendors' response to review

Steven Dubnoff, Circle Systems

[Circle Systems, Conceptual Software, and SPSS *were all helpful in providing software and information related to dm2. Vendors were offered an opportunity to respond to the review, but Conceptual Software and* SPSS*, the developers and marketers, respectively, of* DBMS/COPY*, declined, saying they were satisfied with the review as it stands.* SPSS *added that an* SPSS *and* SAS *Export utility will be available in a future version. Below is the response from Circle Systems, the developer of Stat/Transfer—Ed.]*

Thank you for your careful review of Stat/Transfer. It has already stimulated a minor bug fix and a significant increase in our processing speed. However, users should not bother to update to version 1.4B, since we are working on a major new release. We expect that it will be available in October.

This new version will offer direct support for Paradox, Quattro Pro and Excel. In addition to the menus, it will run in batch mode; it will even automatically generate its own command files. It will have a spiffier user interface and, of course, correct the problems with variable names you mentioned. The price will remain the same, which will make Stat/Transfer the unequivocal "best buy".

Stat/Transfer has not received much of our attention in the past several years. However, we believe that it is essential that vendors of transfer products be independent from the companies that actually develop statistical packages. Now that DBMS/COPY is being marketed exclusively by SPSS, we will devote our energies to making Stat/Transfer the premiere data transfer package. We believe that only independent vendors such as ourselves can be truly responsive to the needs of other developers and of users.

gr6 Lowess smoothing

Patrick Royston, Royal Postgraduate Medical School, London, FAX (011)-44-81-740 3119

The format of the `ksm` command is

`ksm` *yvar xvar* [`if` *exp*] [`in` *range*] [`,` [[`line`] [`weight`] | `lowess`]
[`bwidth(#)`] [`logit`] [`adjust`] [`gen(`*newvar*`)`] [`nograph` | *graph_options*]]

`ksm` carries out unweighted or locally weighted smoothing of *yvar* on *xvar* and displays a smoothed scatterplot of the results. Options are

`line` for running-line least-squares smoothing. Default is running mean.

`weight` to use Cleveland's (1979) tricube weighting function. Default is unweighted.

`lowess`, which is equivalent to specifying "`line weight`" and is equivalent to Cleveland's "lowess" running-line smoother.

`bwidth(#)` to specify the bandwidth. Centered subsets of `bwidth` $\cdot N$ observations are used for calculating the smoothed values for each point in the data except for the end points, where smaller, uncentered subsets are used. The greater the `bwidth`, the greater the smoothing. Default is .8.

`logit` to transform smoothed *yvar* to logits. Predicted values less than .0001 or greater than .9999 are set to $1/N$ and $1-1/N$, respectively, before taking logits.

`adjust` adjusts the mean of the smoothed *yvar* to equal the mean of *yvar* by multiplying by an appropriate factor. This is useful when smoothing binary (0/1) data.

`nograph` to suppress displaying the graph, which is often used with the `gen()` option. Default is to display the graph.

`gen(`*newvar*`)` to create *newvar* containing the smoothed values of *yvar*, in addition to or instead of displaying the graph.

In addition, all the normal `graph` options are valid.

The most important use of `ksm` is to provide lowess (locally weighted regression scatter plot smoothing) as described in Cleveland (1979). The basic idea is to create a new variable (*newvar*) that, for each *yvar* in the data, y_i, contains the corresponding smoothed value. The smoothed values are obtained by running a regression of *yvar* on *xvar* using only the data (x_i, y_i) and a small amount of the data near the point. In lowess, this regression is weighted so that the central point (x_i, y_i) gets the highest weight and points further away from the central point (based on the distance $|x_j - x_i|$) receive less. The estimated regression is then used to predict the smoothed value $\hat{y}_i$ for y_i only. The procedure is repeated to obtain the remaining smoothed values, which means a separate weighted regression is estimated for every point in the data.

Lowess is a desirable smoothing method because of its locality. It tends to follow the data. Polynomial smoothing methods, for instance, are global in that what happens on the extreme left of a scatter plot can affect the fitted values on the extreme right.

The amount of smoothing is affected by the `bwidth` and users are warned to experiment with different values. For instance:

```
. ksm h1 depth, lowess ylab xlab s(0i)                    (Figure 1)
. ksm h1 depth, lowess ylab xlab s(0i) bwidth(.4)         (Figure 2)
```

In Figure 1, the default bandwidth of .8 is used, meaning 80% of the data is used in smoothing each point. In Figure 2, I explicitly specified a bandwidth of .4. Smaller bandwidths, as in Figure 2, follow the original data more closely.

Two `ksm` options are especially useful with binary (0/1) data: `adjust` and `logit`. `adjust` adjusts the resulting curve (by multiplication) so that the mean of the smoothed values is equal to the mean of the unsmoothed values. `logit` specifies the smoothed curve is to be in terms of the log of the odds ratio:

```
. ksm foreign mpg, lowess ylab xlab jitter(5) adjust      (Figure 3)
. ksm foreign mpg, lowess ylab xlab logit yline(0)        (Figure 4)
```

With binary data, if you do not use the `logit` option, it is a good idea to specify `graph`'s `jitter()` option. Since the underlying data (whether a car is manufactured outside the United States in this case) takes on only two values, raw data points are more likely to be on top of each other, thus making it impossible to tell how many points there are. `graph`'s `jitter` option adds some noise to the data to shift the points around. This noise affects only the location of the points on the graph, not the lowess curve. When you do specify the `logit` option, the display of the raw data is suppressed.

`ksm` can be used for other than lowess smoothing. Lowess can be usefully thought of as a combination of two smoothing concepts: the use of predicted values from regression (rather than means) for imputing a smoothed value and the use of the tricube weighting function (as opposed to a constant weighting function). `ksm` allows you to combine these concepts freely. You can use line smoothing without weighting (specify "`line`"), or mean smoothing without weighting (specify no options), or mean smoothing with tricube weighting (specify "`weight`"). Specifying both `weight` and `line` is the same as specifying `lowess`.

Warning: This program is computationally intensive and may therefore take a long time to run on a slow computer. Lowess calculations on 1,000 observations, for instance, require estimating 1,000 regressions. Try a small data set first if in doubt.

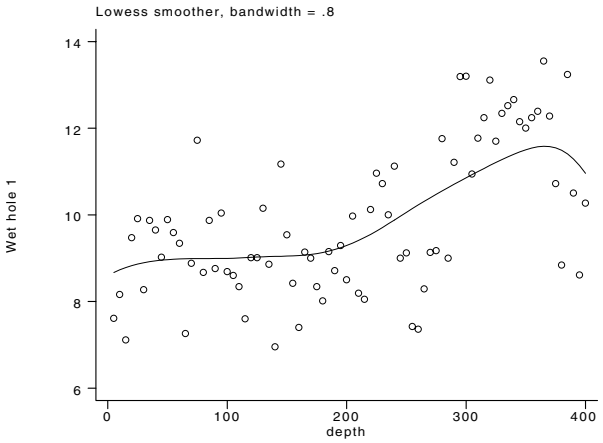


Figure 1

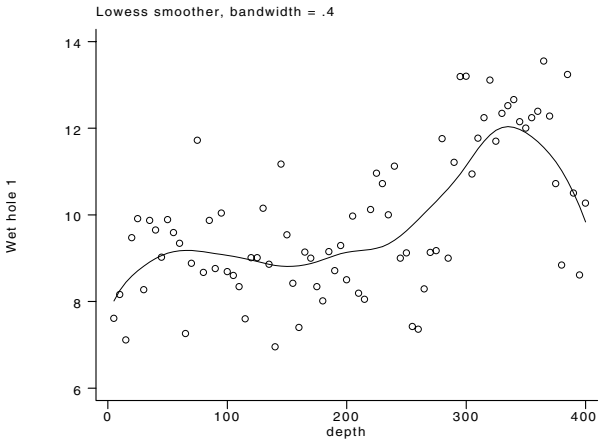


Figure 2

(*Continued on next page*)

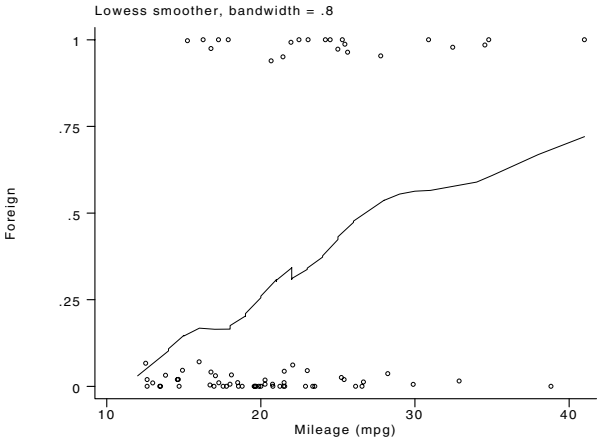


Figure 3

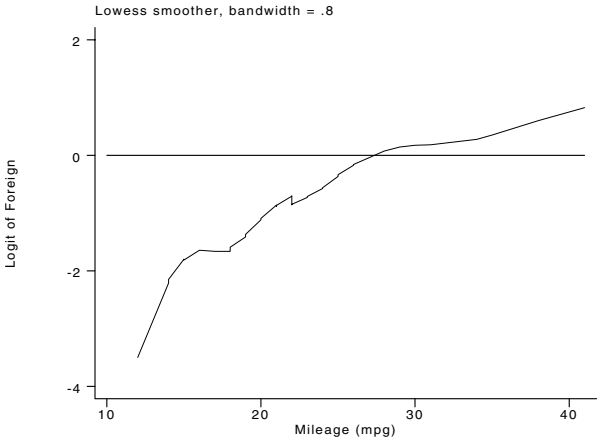


Figure 4

References

Chambers, J. M., W. S. Cleveland, B. Kleiner, and P. A. Tukey. *Graphical Methods for Data Analysis.* Belmont, CA: Wadsworth International Group.

Cleveland, W. S. 1979. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association* 77: 829–836.

Cleveland, W. S. 1985. *The Elements of Graphing Data.* Monterey, CA: Wadsworth Advanced Books and Software.

gr7	Using Stata Graphs in the Windows 3.0 Environment

Joseph Hilbe, Editor, STB, fax 602-860-1446

Many Stata users also compute and program in the Microsoft Windows 3.0 environment. Although it at first appears that using Stata and Windows are two entirely separate domains, this article presents a method to use Windows as (1) a means of Stata graph annotation and (2) a means to incorporate enhanced Stata graphs into any windows based document; e.g., Word for Windows or Excel. You may also simply print the revised graph and paste it into a camera-ready document.

The graph annotation allowed by Windows is rather extensive. You may create textual input as well as graphical lines, circles, ovals, rectangles, polygons, and so forth. You may also vertically and horizontally rotate the image, create almost any color alteration, or use a variety of fonts. In short, by using Windows, you can create a truly customized Stata graph. If you use Stage to first overlay graphs, the effect is even more dramatic.

There may be alternative methods to perform the task at hand, but since Microsoft provides little documentation to assist, I suggest that you follow the procedure outlined prior to attempting deviations. The only caveat is that if you are running Windows 3.0 in enhanced mode, you cannot use the Intercooled version of Stata; use the regular version. However, if you only use the Intercooled version, simply start Windows in the standard or real mode (`win/r` or `win/s`) and Stata will run—although you will notice a reduction in memory. Either way, you will be loading Stata to invoke the previously saved graph you wish to annotate or import into the windows environment.

Begin by typing `win` at the `C:` prompt. After loading Windows, you need to access `Paintbrush`. The default method is to first click on the Accessories icon, click `Restore`, and double-click on the Paintbrush icon. After entering Paintbrush, click the `maximize` button to enlarge the active window (the upper right corner). Perform the following sequence:

1. Click on `control-menu` command button (upper left corner).
2. Click on `Minimize`.
3. Click on `File` in Accessories window.
4. Click on `Close`.
5. Click on `File` in Program Manager window.
6. Click on `Run`.
7. Type and Enter path and `stata.exe` in displayed input box: e.g, `c:\stata\stata.exe`. Press `OK`.
8. Type and Enter Stata command to display a graph in a screen-reduced mode, e.g., "`gr using` *filename*`, mar(40)`". This

reduces the graph image by 40 percent. You can alter as needed.

9. Press `Print Screen` key on keyboard. This loads the screen image into the Windows Clipboard.
10. Simultaneously press the *Alt-Tab* keys on the keyboard. This places you back into Windows with a minimized Stata icon displayed in the lower left-hand corner of the screen.
11. Double-click on the Paintbrush icon.
12. Click on `Restore`.
13. Click `View` on the main menu bar.
14. Click `Palette` to temporarily remove the color selection bar from the lower screen. You may toggle it back again when editing.
15. Click `Edit` and then `Paste`.
16. Click `Scissor` (the rectangle cutout tool on the vertical tool and line bar) and cut out a box covering only that part of the graph you want by dragging cursor from one corner to the other.
17. Click on `Edit` then `Cut`.
18. Click on `File`, `New`. Click `No` button to discard unwanted material.
19. Click on `Edit` then `Paste`. The desired figure is placed in upper left hand corner of the screen.
20. Click on `Pick` from Menu Bar. Select `Inverse` to reverse the black Stata graph background. This will provide a cleaner print.
21. Click on `Scissor` and drag cursor around figure starting from upper left corner. You may then do any of the following:
    a. Annotate the graph using any of the toolbox utilities.
    b. Horizontally or vertically flip the figure, or tilt it within a 180 degree range (from Pick options).
    c. Save the graphic image to a file.
    d. Export the new graph to another Windows program by placing it in the Clipboard, opening the desired program, and pasting it where desired *or* print it with the Windows installed printer.

Experimentation and patience should help you obtain interesting graphic results. Comments, suggestions, and further enhancements are welcome.

os1.1	Update on gphpen and color Postscript use

R. Allan Reese, University of Hull, UK. Fax (011)-44-482-466441

Suggestion

Following the publication of *os1* in STB-1, I received only one suggestion. This pointed out that editing the `ps.plf` file may introduce an end-of-file character that causes the file not to concatenate correctly with the page description. This was not my problem, but is worth pointing out to anyone else who wants to 'adjust' the PostScript preamble. Most editors will add an end-of-file mark by default.

Try loading the file into your preferred editor, add and delete a space and resave the file—call it `ps.one`. Then look at the directory entries. If the new file is one byte larger than `ps.plf`, it will probably no longer work. So after editing the file you need to strip off this character, and can do so by a copy with switches.

```
> copy ps.one/a ps.two/b
```

Version 2 of ps.plf

The interesting news is that version 2 of `ps.plf` as printed in STB-1 *does* work. I fetched a public domain program `GhostScript` version 2.2 to debug the program, and it just ran and produced a color plot on my screen. I recommend `GhostScript` to anyone who wants to play with PostScript and stay green (save paper and not go red in the face waiting for output).

I then tried the same file on a QMS-100 ColorPS printer and got no output with

```
> print testfile.ps
```

but got the correct output immediately with

```
> copy /b testfile.ps lpt1
```

The time saved by using an array instead of multiple `if` tests is probably negligible, but does encourage one to provide extra colors, in particular lower intensities, and to try more complex plots.

os3 Using Intercooled Stata within DOS 5. 0

Joseph Hilbe, Editor, STB, FAX 602-860-1446

I had long been waiting for Microsoft to release DOS 5.0. Several beta testers I knew told me of its ability to place various DOS and TSR files into high memory, thus freeing previously cannibalized conventional memory. It was also claimed to have solved many of the problems which plagued 4.01. Actually, DOS 3.3 was a more bug-free operating system; but I needed 4.01's ability to create partitions larger than 32 megabytes. Having a 660 meg hard drive would have me partitioning *ad naseum*. I also thought that the new DOS text editor, `edit`, might be of some assistance to STB subscribers when submitting inserts. Moreover, I liked the prospect that secondary school students be weaned on an interpreter other than BASICA or GW-BASIC. DOS 5.0 includes QBasic, a structured language similar to QuickBasic with the ability of handling programs up to 160K and with a "look & feel" of the advanced Microsoft and Borland compilers. Hence, I had few reservations about upgrading my system to DOS 5.0.

The installation process left my system's QEMM386 memory manager intact. The new DOS includes its own high memory manager, `himem.sys`, and an expanded memory emulator, `emm386.exe`. QEMM386 and `himem.sys` do not work together. I have been using QEMM386 with Intercooled Stata without a serious problem since the release of the latter but was curious to ascertain whether the new DOS memory managers were better—at least with respect to their interaction with Intercooled Stata. This insert will describe `config.sys` files which have been found to work under each memory manager.

After installation I found that I had no difficulty loading and operating Intercooled Stata (henceforth referred to as simply Stata). The following `config.sys` loads the `mouse`,`ansi`, and `smartdrv` drivers into high memory, loads DOS high, reserves sufficient memory to run non-Windows programs from Windows (hence the need for `smartdrv sys`), and can perform the Stata graph annotations in Windows as described in *gr7*. Very little conventional memory is used to run DOS, thus allowing standard programs more ram; but with a slight loss of allowable extended or expanded memory.

```
DEVICE=C:\SETVER.EXE
DEVICE=C:\QEMM\QEMM386.SYS RAM
DEVICE=C:\QEMM\LOADHI.SYS /r:2 C:\MOUSE.SYS
DEVICE=C:\QEMM\LOADHI.SYS /r:3 C:DOS\ANSI.SYS
STACKS=0,0
BREAK=ON
BUFFERS=30
FILES=40
SHELL=C:\DOS\COMMAND.COM C:\DOS\ /E:256 /p
DEVICE=C:\QEMM\LOADHI.SYS /r:1 C:\WINDOWS\smartdrv.sys 1024 512
DOS=HIGH
```

Running Stata without QEMM386 is a bit more complex. The problem seems most apparent when one loads DOS and various device drivers into high memory. The following `config.sys` seems to work for 386 and 486 computers with 4 or more megabytes of ram. I have not tested it on machines with less.

```
DEVICE=C:\DOS\SETVER
DEVICE=C:\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE 2000 RAM
DOS=HIGH
DOS=UMB
DEVICEHIGH=C:\DOS\ANSI.SYS
DEVICEHIGH=C:\MOUSE.SYS
DEVICE=C:\WINDOWS\SMARTDRV.SYS 1024 512
STACKS=0,0
BREAK=ON
BUFFERS=30
FILES=40
SHELL=C:\DOS\COMMAND.COM C:\DOS\ /E:256 /p
```

The above example files can be altered to suit individual requirements. However, if you are not using QEMM386, you must incorporate the following lines into the `config.sys` file in order to more closely emulate QEMM386 capabilities.

```
DEVICE= path:\HIMEM.SYS
DEVICE= path:\EMM386.EXE   number RAM
DOS=HIGH
DOS=UMB
```

Experimentation will provide you with the optimal amount of expanded memory emulation needed for your applications. Each system will require different settings; and settings will vary depending on the applications you wish to install. You must be particularly careful when using Windows to load non-Windows programs. I have not been able to load Intercooled Stata from Windows running in enhanced mode using either QEMM386 or `himem.sys`. I will admit, however, that I have not tried very hard to do so. I should be most interested in learning of successful attempts. I also believe that STB readers will be interested in hearing about alternative methods of configuring DOS to enhance Stata. Please forward them to me for inclusion in future issues.

qs4	Request for additional smoothers

Isaias H. S. Ugarte, Universidad National Autonoma de Mexico, Mexico D. F. Mexico

I should like to inquire if any Stata users have created ado files related to the following smoothers: 4253H, twice; 3RSSH, twice; 43R5R2H, twice; 3RSSH; 53H, twice. I am particularly interested in the first two. References can be found in the work of Paul F. Velleman and John W. Tukey. Forward any information to the STB Editor or to me at Universidad 2014 Bolivia 13, Copilco Universidad Coyoacan 04360, Mexico D. F. Mexico.

sbe3	Biomedical analysis with Stata: radioimmunoassay calculations

Paul J. Geiger, USC School of Medicine, pgeiger@uscvm

Radioimmunoassay (RIA) is a widely used technique in biomedical laboratories. Sundqvist et al. (1989) used it for a testosterone assay involving ^{3}H-labeled ('hot') testosterone as antigen competing with unlabeled ('cold') testosterone in the test sample for the binding sites of a polyclonal antibody against testosterone-BSA (bovine serum albumin) conjugate. The free ('hot') antigen is bound to dextran-coated charcoal and separated from the antibody-bound antigen by centrifugation. The radioactivity, CPM (counts per minute), of the antibody-bound fraction is determined in a scintillation counter. A complete discussion of the method as well as related techniques, its design and caveats, is found in Chard (1990).

The analysis of raw RIA data is often done by the so-called logit-log plot developed by Rodbard and Lewald (1970). The method has received much attention and many programs have been written to apply it. The recent book by Chard (1990) and a review by Rodbard et al. (1987) are given in the references. The logit transformation is

$$\frac{\text{logit}(Y) = \log(Y}{1 - Y)}$$

Variable Y is the fraction of bound standard CPM corrected for non-specific binding. The `logitY` is the natural log of the ratio shown above and this is plotted against log(base 10) of the concentration (`logconc`) in picograms per milliliter (pg/ml).

Replicates of standards as well as unknowns are essential in this procedure, triplicate or even quadruplicate samples for standards and at least duplicates for each level of dilution of the unknowns.

Linear regression is performed on `logitY` vs. `logconc`. The resulting coefficient (`_b[_logconc]`) and constant (`_b[_cons]`) are then used together with the computed logit of the CPM of the unknowns (`logitS`) in the reverse transformation to calculate the pg/ml for the unknown or test samples. Final answers are computed using the appropriate volume and dilution factors for each sample. Confidence limits for the answers are difficult to obtain because of the heteroscedasticity of the logit transformed variables.

In order to carry out the above procedure Sundqvist et al. (1989) designed a spreadsheet with separate templates for data entering, macros and formulas, etc. The standards entry table is limited to 9 standards in triplicate but could accommodate more with adjustments to the cells in the template. Means are calculated for CPM values, both standards and unknowns, before they are transformed. Provision is made to view the logit-log plot before regression and the go-ahead for analysis of unknowns is given after regression produces a satisfactory correlation coefficient, a value > 0.99, for the standards. The results for the unknowns are shown in a separate table along with the volumes and dilution factors. The same table serves as the entry template for the raw CPM of the unknowns.

Analysis of Sundqvist's data has been carried out with Stata by means of simple do-files using Stata commands and language. Descriptions of each of the steps are provided. The intent is to illustrate the power of Stata for such biomedical work, as Spreadsheets have limited mathematical and statistical capability.

A "template" includes all of the relevant data for the analysis:

```
dictionary {
B                ``CPM of total binding´´
b100             ``100 percent binding´´"
NSB              ``Nonspecific binding´´
stds_pg          ``Standards in pg/ml´´
cpm              ``CPM of standards´´
smpl_num         ``Sample ID number´´
Scpm             ``Sample cpms, raw´´
Vol_ml           ``Volume of sample, ml´´
Dil_fact         ``Dilution factor´´
}
5530 1580 35   2.5  1466 11772     773  .025 .04
5522 1598 38   2.5  1463 11772     770  .025 .04
5541 1576 33   2.5  1457 11772     571  .05  .04
.    .    .    5    1431 11772     566  .05  .04
.    .    .    5    1445 11773     879  .025 .08
.    .    .    5    1430 11773     631  .05  .8
.    .    .    10   1296 11774     920  .025 .04
.    .    .    10   1301 11774     928  .025 .04
.    .    .    10   1289 11774     715  .05  .04
.    .    .    25   989  11774     700  .05  .04
.    .    .    25   980  11775     777  .025 .04
.    .    .    25   982  11775     765  .025 .04
.    .    .    50   684  11775     577  .05  .04
.    .    .    50   685  11775     568  .05  .04
.    .    .    50   680  11776     876  .025 .8
.    .    .    100  487  11776     633  .05  .8
.    .    .    100  490  11777     915  .025 .04
.    .    .    100  489  11777     716  .05  .04
.    .    .    150  378  11778     838  .025 .8
.    .    .    150  380  11778     578  .05  .8
.    .    .    150  377  11779     833  .025 .8
.    .    .    250  260  11779     828  .025 .8
.    .    .    250  256  11779     601  .05  .8
.    .    .    250  264  11779     615  .05  .8
.    .    .    500  158  11780     989  .025 .04
.    .    .    500  159  11780     1001 .025 .04
.    .    .    500  150  11780     755  .05  .04
.    .    .    .    .    11780     786  .05  .04
```

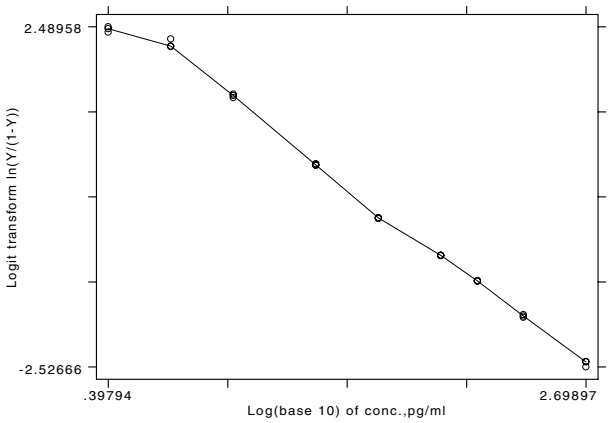
This template is simply a Stata dictionary file (included as `ria.dct` on the STB disk along with the other relevant do-files) ready to import with the command "`infile using ria.dct`". The investigator can store the template for use with different values in another set of experiments, replacing only the numbers with his own.

The template is created with a wordprocessor by simply typing in the variable names and labels using the Stata format for creating a data set with a dictionary and saving it as an ASCII file. This method is easier to use than the `input` command as the variables are then already labeled. Alternatively, since almost every laboratory has a microcomputer with spreadsheet these days, they can be used to make the layout of variables and values. The resulting file can then be imported into Stata with the program Stat/Transfer or DMBS/COPY, or saved in ASCII format imported as raw data. *[See os2 in this issue for a review of data transfer programs—Ed.]*

Once the RIA file is imported the first do-file, `ria.do`, is run by typing "`run ria.do`". The file has been written in simple steps with comments and can be printed out in order to check or change variable names and to see how the formulas have been applied to the RIA analysis. The step generating variable `B_To` has been included to show the fraction of the total bound.

A value of about 0.3 or more is desirable and Chard (1990) says 0.5 or 50% should be sought in designing an RIA. Sundqvist et al. got 28.59%.

As the calculations proceed, a graph is displayed in order to see the shape of the logit-log curve:



The graph has been kept simple by allowing Stata graphics to size and number the axes with the built-in program. With this particular set of data the curve at the upper left shows the lack of reliability (expected) at the low end of concentration values. Regression is then performed of `logitY` on `logconc`. The next graph shown is the plot of the regression line (`hat`) fitted to the experimental points. Although it has not been done here, the `graph` commands can be altered to save the graphs separately if desired by including the `saving(`*filename*`)` option.

The last to appear on the screen is the regression table so that the value of R^2 can be checked:

```
Source   |       SS         df       MS        Number of obs =      27
---------+------------------------------       F(  1,    25) = 4433.85
Model    |  75.0533305       1  75.0533305      Prob >F       =  0.0000
Residual |  .423184083      25  .016927363      R-square      =  0.9944
---------+------------------------------       Adj R-square  =  0.9942
   Total |  75.4765146      26  2.90294287      Root MSE      =  .13011

Variable |  Coefficient   Std. Error     t    Prob > |t|        Mean
---------+--------------------------------------------------------------
  logitY |                                                    -.0111083
---------+--------------------------------------------------------------
 logconc |    -2.236135     .0335821  -66.587    0.000      1.607425
   _cons |      3.58331     .0595051   60.219    0.000             1
---------+--------------------------------------------------------------
```

The R^2 is greater than 0.99 as desired.

The file `smplria.do` is then run to calculate values and answers for the unknowns based on the regression from `ria.do`. Results are presented including the variables to identify the sample number, the actual pg/ml calculated, and the final answers corrected for volume:

```
     (Sample   (pg/ml)   (pg/ml    pg/ml
     ID)       pg_ml     corr.)    Ref.[1]
     smpl_num            answer    means
-----------------------------------------------
     11772     44.18   1767.13
     11772     44.53   1781.31     1859.11
     11772     77.21   1544.25
     11772     78.36   1567.16     1634.48
     11773     33.31   1332.42
     11773     65.00   1300.03     1389.57
     11774     29.83   1193.03
     11774     29.18   1167.37     1227.63
     11774     51.61   1032.11
     11774     53.74   1074.86     1105.71
     11775     43.71   1748.40
     11775     45.13   1805.22     1861.62
     11775     75.87   1517.34
     11775     77.90   1557.94     1615.47
     11776     33.58   1343.17     1401.00
     11776     64.64   1292.76     1358.63
     11777     30.23   1209.31     1258.64
     11777     51.47   1029.33     1080.38
     11778     37.16   1486.53     1553.45
     11778     75.65   1512.91     1589.77
```

```
11779     37.66   1506.45
11779     38.17   1526.62     1585.31
11779     70.78   1415.52
11779     68.01   1360.10     1458.23
11780     24.66    986.39
11780     23.84    953.58     1004.17
11780     46.35    927.03
11780     42.68    853.50     932.07
```

The results can be seen again at any time by using the `describe` command followed by typing "`list` *thisvar thatvar othervar*" to choose those variables one wants to see. Figure 4 illustrates the results and permits readers to compare with Sundqvist's values.

The now complete `ria.dta` file must be titled with the operator's name, date and other experimental information using the command "`label data "mydata, rats, testosterone, date, etc."`". We have found file names such as `910807_a.dta` convenient, with `a`, `b`, `c` or `1`, `2`, `3` keyed to the laboratory notebook, date, and experiment.

Notes

1. One peculiarity has appeared in that the final values calculated in the present work are found to be low by about 50 to 80 pg/ml compared to the original paper. The answer seems to be that Sundqvist et al. did not subtract the `N` or non-specific binding CPM in calculating their logit transformation. This step is essential (Rodbard et al. (1987)). If `N` is eliminated from the do-file formulas presented in this communication, virtually exact correspondence between answers is obtained. With all due respect, we should note that the NSB is only 0.64% of the total and only 2.23% of the 100% bound and may have been left out intentionally since it makes little practical difference.
2. If one desires a somewhat better fit, the command `rreg` can be used in the `ria.do` file. This has been tried, but with the sample replicate values as good as they are, seems to make little difference also. Besides, as I understand it, using robust regression on fewer than about 30 values, that is, small sample statistics, is incorrect. In biological laboratory experiments very often we can only afford relatively small samples owing to both monetary and physical constraints.

References

Chard, T. 1990. An Introduction to Radioimmunoassay and Related Techniques, vol. 6, part 2 of *Laboratory Techniques in Biochemistry and Molecular Biology*, ed. R. H. Burdon and P. H. van Knippenberg. New York: Elsevier.

Rodbard, D. and J. E. Lewald. 1970. Computer Analysis of Radioligand Assay and Radioimmunoassay Data. *Acta Endocr. Suppl.* 147: 79–103.

Rodbard, D., et al. 1987. Statistical Aspects of Radioimmunoassay in *Radioimmunoassay in Basic and Clinical Pharmacology*, vol. 82 of *Handb. Exp. Pharm.*, ed. C. Patrono and B. A. Peskar, chapter 8. New York: Springer-Verlag.

Sundqvist, C. et al. 1989. A Radioimmunoassay Program for Lotus 1-2-3, *Comput. Biol. Med.* 19: 145–150.

sed4	Resistant normality check and outlier identification

Lawrence C. Hamilton, Dept of Sociology, Univ. of New Hampshire

A single outlier can dramatically inflate the usual skewness and kurtosis statistics, which depend on third and fourth powers of deviations from the mean. Exploratory data analysis (EDA) enthusiasts often prefer to work with more resistant statistics for describing distributional shape (see Deleon, 1991, and the sources he cites). Order statistics, including median and quartiles, combine high resistance to outliers with easy calculation and interpretation. For example, comparing mean with median diagnoses overall skew:

mean $>$ median	positive skew
mean $=$ median	symmetry
mean $<$ median	negative skew

The greater the mean–median difference, the less plausible the mean as a summary of the distribution's "center."

The median could be described as a "50% trimmed mean": the average disregarding both the top 50% and the bottom 50% of the data. A less radical, but still resistant, summary measure is the 10% trimmed mean: the average of cases between 10th and 90th percentiles. Trimmed means are simple robust estimators, retaining (unlike the median) much of the normal-distribution efficiency of a mean, but performing better than means with heavy-tailed distributions. In a symmetrical distribution, the trimmed mean equals the median and mean.

If the distribution appears roughly symmetrical, we might go a step further to make a simple normality check involving the pseudo-standard deviation (PSD):

standard deviation $>$ PSD	heavier-than-normal tails
standard deviation $=$ PSD	normal tails
standard deviation $<$ PSD	lighter-than-normal tails

The PSD is defined as IQR$/1.349$, where IQR is the interquartile range (IQR $=$ Q3 $-$ Q1, or 75th percentile minus 25th percentile). In a normal distribution, standard deviation $=$ PSD. Since PSD depends on spread in the middle 50% of a distribution, ignoring the tails, it is unaffected by outliers. The standard deviation, in contrast, has even less resistance than the mean, because it depends on squared deviations. Standard deviation/PSD comparisons are less informative if the distribution is very skewed, because (a) the skew is evidence against normality already, and (b) skewed distributions typically have one lighter and one heavier tail.

One of EDA's most successful innovations, the boxplot, graphically displays median, IQR, and outliers. Stata boxplots identify as outliers any data points more than 1.5IQR below the first quartile or 1.5IQR above the third quartile. The cutoffs Q1 $-$ 1.5IQR and Q3 $+$ 1.5IQR are called inner fences. Values beyond the inner fences may be no cause for alarm; they make up about 0.7% of a normal population.

Other boxplot implementations distinguish between "mild" and "severe" outliers. The usual definitions are

$$x \text{ is a mild outlier if } \quad \text{Q1} - 3\text{IQR} \le x < \text{Q1} - 1.5\text{IQR} \quad \text{or} \quad \text{Q3} + 1.5\text{IQR} < x \le \text{Q3} + 3\text{IQR}$$

$$x \text{ is a severe outlier if } \quad x < \text{Q1} - 3\text{IQR} \quad \text{or} \quad x > \text{Q3} + 3\text{IQR}$$

The cutoffs Q1 $-$ 3IQR and Q3 $+$ 3IQR are called outer fences; severe outliers fall beyond the outer fences. Severe outliers comprise about two per million (.0002%) of a normal population. In samples, they lie far enough out to have substantial effects on means, standard deviations, and other classical statistics.

Due to sampling variation in quartiles, outliers appear more often in small samples than one might expect from their population proportions. Monte Carlo simulations by Hoaglin, Iglewicz, and Tukey (1986) obtained these results:

Percentage of outliers in random samples from normal population

n	any outliers	severe
10	2.83%	.362%
20	1.66	.074
50	1.15	.011
100	.95	.002
200	.79	.001
300	.75	.001
infinite	.70	.0002

They employed a different approximation for sample quartiles than the one Stata uses, but this should not affect the general pattern. (For a discussion of quartile approximations see Frigge, Hoaglin, and Iglewicz, 1989. Stata's boxplots use their definition 5, as do SPSS and StatGraphics. Minitab, SAS, and Systat use other definitions.)

Could the sample at hand, outliers and all, plausibly have come from a normal population? Hoaglin et al. report the following percentages of samples (from a normal population) containing outliers:

Percentage of samples containing outliers

n	any outliers	severe
10	20.3%	2.9%
20	23.2	1.2
50	36.4	0.5
100	52.9	0.2
200	72.9	0.3
300	85.2	0.2
infinite	100.0	100.0

Note that the percentage of normal samples containing severe outliers declines as sample size increases from small to moderate. In much larger samples, the percentage with severe outliers increases again, towards 100% for infinite-size samples. Judging from this table, the presence of any severe outliers in a real-data sample of $n = 10$ to at least 300 should be sufficient evidence to reject normality at a 5% significance level. Mild outliers, on the other hand, appear common in samples of any size.

Severe outliers in samples thus should often cast doubt on normality assumptions. Furthermore, such outliers represent the kind of nonnormality most hazardous to classical statistical techniques. Outliers may be interesting for substantive as well as statistical reasons; they represent cases much different from most of the data. Outlier labeling has a less obvious use in evaluating regression diagnostic statistics such as hat diagonals (leverage), Cook's D, or DFBETAS. A case with a severe-outlier Cook's D, for instance, represents a "severe influence outlier": it is much more influential than most other cases.

Why not stick with traditional outlier-detection methods, based on standard deviations from the mean? Since extreme values pull the mean and inflate the standard deviation, even a severe outlier may not be many standard deviations from the mean—a problem called masking. Resistant outlier detection, on the other hand, does not suffer from masking—extreme values cannot much affect the criteria.

`iqr.ado` prints these univariate statistics: mean, median, and 10% trimmed mean; standard deviation, PSD, and IQR; inner and outer fences; and the number and percentage of mild and severe outliers. The program may be particularly useful in teaching, because it allows students to perform simple normality and outlier checks as a routine part of data analysis (instead of routinely assuming normality, as in many old-fashioned texts). Order statistics require less explaining than quantile-normal plots or formal normality tests, yet may work as well in detecting serious nonnormality and outliers. (See Gould, 1991, for an unencouraging report on some formal normality tests.) Thus `iqr` provides an EDA-flavored supplement to Stata's `summarize, detail` command.

The following example uses data from the Boston Globe regarding average coliform bacteria counts at 21 Boston-area beaches during the summer of 1987.

```
. list beach bacteria
        beach      bacteria
 1.    Yirrel          8
 2.     Short          8
 3.  Houghton          9
 4.     Sandy         10
 5.    Stacey         10
 6.  Nantaske         12
 7.  Winthrop         13
 8.    Revere         13
 9.    Lovell         13
10.    Nahant         14
11.    Pearce         15
12.    Malibu         16
13.    Peckem         18
14.  Swampsco         21
15.     Kings         22
16.      Lynn         22
17.  Pleasure         30
18.  Constitu         35
19.    Carson         45
20.    Tenean         52
21.  Wollasto         88

. iqr bacteria
   mean=  22.57          std.dev.=   19.2          (n= 21)
 median=     15    pseudo std.dev.=  7.413        (IQR= 10)
10 trim=   17.6

                                          low         high
                                  -------------------------
                      inner fences        -3           37
                   # mild outliers     0           2
                   % mild outliers     0.00%       9.52%
                         outer fences      -18          52
                   # severe outliers   0           1
                   % severe outliers   0.00%       4.76%
```

We see a positively skewed distribution with two mild and one severe outliers. Experimenting with Tukey's ladder of powers, `iqr` will confirm that logarithms of bacteria counts remain positively skewed, with one mild outlier. Negative reciprocal roots,

```
. generate nrrbact=-(bacteria^-.5)
```

are more nearly symmetrical, with no outliers. Alternatively, our interest might focus on the outliers themselves. Why are these three beaches so polluted? For publication purposes, the outlier information from iqr also assists Stage enhancement of basic Stata boxplots. In Figure 1, I left the mild outliers as circles but changed the single severe outlier (Wollaston Beach) to a plus sign within a square. This follows the graphical conventions of other boxplot programs that visually distinguish mild from severe outliers.

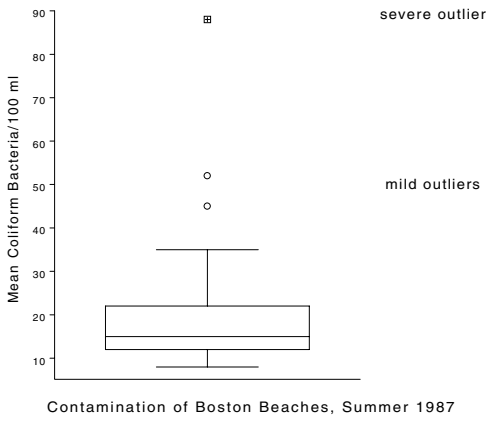


figure 1

References

Deleon, R. E. 1991. sed1: Stata and the four r's of EDA, *Stata Technical Bulletin* 1: 13–17.

Frigge, M., D. C. Hoaglin, and B. Iglewicz. 1989. Some implementations of the boxplot. *The American Statistician* 43(1): 50–54.

Gould, W. 1991. sg3: Skewness and kurtosis tests of normality. *Stata Technical Bulletin* 1: 20–21.

Hoaglin, D. C., B. Iglewicz, and J. W. Tukey. 1986. Performance of some resistant rules for outlier labeling. *Journal of the American Statistical Association* 81(396): 991–999.

sed5 | Enhancement of the Stata collapse command

Paul Banens, CQM, Netherlands, FAX (0)40-758712

The CRC ado-file `collapse` is a useful tool in handling data. In effect it allows the user to transform a data set in memory to one consisting of only means, counts, or medians by sub-group. However, the original data set is always erased from active memory. The `stats` ado-file *[provided on the* STB-3 *disk—Ed.]* provides more options, including a default that does not erase the original data set from memory. It also allows the user to select more than one summary statistic. The syntax of the `stats` command is

`stats` *varlist* [`if` *exp*] [`in` *range*] [`,` [*type* [*type*...]]
[`by(`*varlist2*`)`] [`nodescr`] [`collapse` [`nowarning`] [`keep(`*varlist3*`)`]]]

where *type* is one of

type	Meaning	prefix
`count`	number of non-missing observations	`CT`
`mean`	mean	`MN`
`median`	median	`MD`
`var`	variance	`VR`
`sdev`	standard deviation	`SD`
`min`	minimum	`MI`
`max`	maximum	`MA`
`range`	range (max–min)	`RG`
`sum`	sum of observations	`SM`
`lf`	25th percentile	`LF`
`uf`	75th percentile	`UF`
`df`	interquartile range (75th–25th percentile)	`DF`
`perc(#)`	percentile indicated by #, 0–100	`PT`

`stats` adds variables to the data set in memory containing statistics as specified by one or more *type*'s. The default type is `mean`. If `by(`*varlist2*`)` is specified , the statistics will be calculated for each set of values of *varlist2*. The new variables have the same names as the old ones with a two-letter prefix, shown above, indicating the type of the statistic.

If the option `collapse` is used, the dataset will be collapsed to one observation for each set of values of *varlist2*. Only the added statistics and the variables in *varlist2* and *varlist3* will remain. `nowarning` suppresses a warning against destroying the data set. The description of the output data set can be suppressed by `nodescr`.

The power of `stats` is best described by listing the differences with `collapse`:

1. `stats` adds the specified statistics as new variables to the complete data set in memory. Only when the option `collapse` is used will the data set be collapsed into one observation per sub-group.
2. `stats` can create more statistics at the same time—up to 13.
3. `stats` can work on the entire data set without a subgrouping `by()`.
4. `stats` can keep variables of the original data set when using the `collapse` option, although this retains only the last observation in each sub-group.
5. `stats` handles the `if` and `in` specification in a different manner than does `collapse`. `stats` selects first and then sorts according to the (optional) by-group. `collapse`, on the other hand, sorts first and then selects according to `if` and `in`. The latter can prove quite dangerous.
6. `stats` provide the user with many summary statistical options, including a free choice point percentile.

Examples

Typing "`stats age income`" adds the variables `MNage` and `MNincome` to the dataset , containing the means (default) of `age` and `income`.

Typing "`stats age income, by(region) mean`" adds the variables `MNage` and `MNincome` containing the means of `age` and `income` by `region`.

Typing "`stats age income, by(region) mean perc(33) collapse`" produces a new dataset of `region`, `MNage`, `PTage`, `MNincome`, and `PTincome` containing the mean and 33rd percentile per `region`, one observation per `region`. Since converting the data results in the loss of the data in memory, `stats` warns you and asks if you really want to continue. `nowarning` suppresses the warning and the subsequent question and answer.

sg1.1 Correction to the nonlinear regression program

Joseph Hilbe, Editor, STB, FAX 602-860-1446

Dr. Paul Geiger has pointed out to me that the output produced by the non-linear regression program `nonlin.ado` incorrectly displays the one less than the number of iterations rather than the number of iterations. The problem is fixed and the revised program is on the STB-3 disk.

sg3.2 Shapiro–Wilk and Shapiro–Francia tests

Patrick Royston, Royal Postgraduate Medical School, London, FAX (011)-44-81-740 3119

As promised last time in *sg3.1*, I now supply as ado-files the alternative Shapiro–Wilk W and Shapiro–Francia W' tests. The syntaxes are

`swilk` *varlist* [`if` *exp*] [`in` *range*] [`, lnnormal`]
`sfrancia` *varlist* [`if` *exp*] [`in` *range*]

`swilk` performs the Shapiro–Wilk test, testing either for normality or, if the `lnnormal` option is specified, log-normality, meaning $\log(X-k)$ is tested for normality, where k is estimated from the data as the value which makes the skewness coefficient $\sqrt{b_1}$ zero. `sfrancia` performs the Shapiro–Francia test.

```
. swilk lchol
                    Shapiro-Wilk W test for normal data
 Variable |    Obs           W         V          z    Pr>z
 ---------+-----------------------------------------------------
    lchol |     80     0.97236     1.259      0.603  0.27312
. sfrancia lchol
                Shapiro-Francia W' test for normal data
 Variable |    Obs          W'        V'          z    Pr>z
 ---------+-----------------------------------------------------
    lchol |     80     0.98996     0.757     -0.566  0.71431
```

The tests report V and V' in addition to W and W', which are more appealing indexes for departure from normality than W and W'. There is no different or additional information in V (V') than in W (W'), one is simply a transform of the other. The median values of V and V' are 1 for samples from normal populations. Large values indicate non-normality. The 95% critical poits of V (V'), which depend on the sample size, are between 1.2 and 2.4 (2.0 and 2.8). For more information, see P. Royston, "Estimating Departure from Normality," *Statistics in Medicine*, 10, 1283–1293, 1991.

sg3.3	Comment on tests of normality

Ralph B. D'Agostino, Albert J. Belanger, and Ralph B. D'Agostino Jr., Boston University

[The following insert is in response to sg3.1 by Patrick Royston entitled Tests for Departure from Normality, July 1991.—Ed.]

We read with interest the recent note by Royston on the D'Agostino–Pearson K-square test for normality (Pearson, D'Agostino and Bowman, 1977) and have several comments. First, we did not overlook the dependency of the skewness and kurtosis statistics. We clearly stated that the K-square statistic "has approximately a chi-squared distribution" (D'Agostino, Belanger and D'Agostino Jr. 1990). We did not say it was exactly chi-squared. The normal plots of Royston clearly demonstrate that the statistic is approximately chi-squared. Also, our article makes it clear that the K-square test and the individual tests for skewness and kurtosis are approximate tests. The word approximate or approximately is used six times in describing the tests so as not to mislead.

Second, our simulations to evaluate the rejection probabilities show the K-square tests, as given in the 1990 article, do have actual levels of significance close to the nominal levels. However, even if we accept Royston's simulation results, we do not find it bothersome to have a nominal level of 0.05 and an actual level less than 0.06. For any realistic application a difference of this size is usually acceptable. To insure that the actual and nominal levels are equal the user can perform the test as given in the 1977 article. The approximations employed in the 1990 version do not change the actual levels in any serious fashion.

Third, the main objective of our 1990 article was to replace the Kolmogorov test with more powerful and informative tests, especially for $n > 50$ where the available computer software did not have the Shapiro–Wilk test. Royston now suggests the use of his version of this and the W' test. This is fine, except that as sample sizes increase, as any applied researcher knows, these tests will reject the null hypothesis of normality. The question then becomes, why? The next question usually is does it matter? At this point the use of skewness and kurtosis statistics and the normal probability plots, as recommended in our 1990 article, are needed to guide us. These not only tell one to reject normality, but why (e.g., skewed data or kurtosis greater than normal kurtosis). Further they can help us judge if our later inferences will be affected by the nonnormality. The W test tells us to reject, it tells us nothing more. Our opinion is that if one is going to be led to the skewness and kurtosis statistics, he/she should start with them.

References

D'Agostino, R. B., A. Belanger and R. B. D'Agostino, Jr. 1990. A suggestion for using powerful and informative tests of normality. *American Statistician* 44(4): 316–321.

Pearson, E. S., R. B. D'Agostino and K. O. Bowman. 1977. Tests of departure from normality: comparison of powers. *Biometrika* 64: 231–246.

Royston, J. P. 1991. sg3.1: Tests for departure from normality. *Stata Technical Bulletin* 2: 16–17.

sg3.4	Summary of tests of normality

William Gould and William Rogers, CRC, FAX 213-393-7551

In this insert, we update the tables presented in *sg3* to include the Shapiro–Wilk and Shapiro–Francia tests suggested by Royston in *sg3.1* and submitted in *sg3.3*, and we provide a response to Royston's comments concerning the various tests of normality. To begin, the updated table is

True Distribution	Test	1%	5%	10%	True Distribution	1%	5%	10%
Normal					Contaminated Normal			
	sktest	.024	.054	.084		.967	.971	.973
	Bera-Jarque	.020	.044	.070		.967	.970	.972
	D'Agostino	.018	.059	.100		.965	.970	.973
	swilk	.010	.048	.098		.949	.956	.959
	sfrancia	.010	.057	.108		.963	.970	.973
Uniform					Long-tail Normal			
	sktest	.007	.652	.938		.130	.216	.283
	Bera-Jarque	.002	.567	.914		.118	.197	.259
	D'Agostino	.985	.997	.999		.081	.179	.263
	swilk	.998	1.000	1.000		.027	.071	.119
	sfrancia	.767	.970	.993		.089	.229	.343
t(5)					t(20)			
	sktest	.549	.641	.693		.096	.151	.198
	Bera-Jarque	.535	.624	.677		.088	.135	.179
	D'Agostino	.453	.595	.673		.069	.137	.197
	swilk	.239	.334	.394		.018	.055	.098
	sfrancia	.466	.629	.712		.055	.142	.215
chi2(5)					chi2(10)			
	sktest	.926	.985	.995		.667	.834	.906
	Bera-Jarque	.892	.972	.992		.609	.786	.873
	D'Agostino	.883	.977	.995		.606	.806	.895
	swilk	.988	.998	1.000		.753	.891	.940
	sfrancia	.974	.996	.998		.711	.880	.933

To refresh your memory, the numbers reported are the fraction of samples that are rejected at the indicated significance level. We performed tests by drawing 10,000 samples, each of size 100, from the indicated distribution. Each sample was then run through each test and the test statistic recorded. (Thus, each test was run on exactly the same sample.) We previously made the following observations:

1. The results under "Uniform" provide the most striking evidence in favor of the D'Agostino test. We would now add the Shapiro–Wilk test to the favorable category and, except at the 1% level, the Shapiro–Francia test.
2. For the t(5) distribution, which we feel is a reasonable real-life distribution, `sktest` outperforms D'Agostino. We would now add that the Shapiro–Francia test performs almost equally well as `sktest`, but that the Shapiro–Wilk test performs miserably in this case.
3. We noted similar results for the "Long-tail Normal" as for t(5) and we would now add the same comment with respect to the Shapiro–Francia and Shapiro–Wilk tests. Shapiro–Francia does well but Shapiro–Wilk does not.

In summary, then, the tables show that the Shapiro–Francia test performs excellently, as Royston claimed. Nevertheless, the Shapiro–Wilk and Shapiro–Francia tests are based on rank statistics and we were concerned that, as with other of such tests, they might not deal with aggregate (or tied) data well. To demonstrate the potential problem to ourselves, we begin with the following example:

```
. set seed 1001
. set obs 1000
obs was 0, now 1000
. gen z = invnorm(uniform())
. sktest z
               Skewness/Kurtosis tests for Normality
                                                 ------- joint -------
 Variable |  Pr(Skewness)   Pr(Kurtosis)      Chi-sq(2)  Pr(Chi-sq)
----------+-----------------------------------------------------------
        z |      0.951          0.898            0.02       0.990
. swilk z
                  Shapiro-Wilk W test for normal data
 Variable |    Obs           W         V          z     Pr>z
 ---------+-------------------------------------------------
        z |   1000     0.98921     0.808     -1.148  0.87446
. sfrancia z
              Shapiro-Francia W´ test for normal data
 Variable |    Obs          W´        V´          z     Pr>z
 ---------+-------------------------------------------------
        z |   1000     0.99880     0.804     -0.523  0.69963
```

We created 1,000 normally distributed random numbers, without aggregation, and ran the data through all three tests. All are

in agreement—none reject that the data is from a normal distribution. We then aggregated the data by rounding it into units of halves:

```
. gen z2 = int(z*2+sign(z)/2)/2
. tab z2, plot
          z2|      Freq.
------------+-----------+-----------------------------------------------------
       -3.5 |          1 |
         -3 |          1 |
       -2.5 |          8 |**
         -2 |         27 |*******
       -1.5 |         69 |******************
         -1 |        118 |*******************************
       -.5  |        193 |**************************************************
          0 |        197 |***************************************************
         .5 |        177 |**********************************************
          1 |        109 |****************************
        1.5 |         60 |***************
          2 |         34 |*********
        2.5 |          4 |*
          3 |          1 |
        3.5 |          1 |
------------+-----------+-----------------------------------------------------
      Total |       1000
```

The 1,000 observations now take on only fifteen distinct values, but those fifteen values are roughly normally distributed. (Given the aggregation, the data obviously cannot still be normal, but we will assert that the data is "normal enough" for most applications—it has no dangling tails nor is it skewed, properties that would cause many statistical procedures problems.) Let us now test this aggregated data for normality:

```
. sktest z2
                Skewness/Kurtosis tests for Normality
                                                 ------- joint -------
 Variable |  Pr(Skewness)   Pr(Kurtosis)      Chi-sq(2)  Pr(Chi-sq)
----------+-----------------------------------------------------------
       z2 |      0.670          0.907             0.19       0.907
. swilk z2
                    Shapiro-Wilk W test for normal data
 Variable |    Obs           W         V          z     Pr>z
 ---------+-------------------------------------------------
       z2 |   1000     0.96940     2.290      8.217  0.00000
. sfrancia z2
                Shapiro-Francia W´ test for normal data
 Variable |    Obs          W´        V´          z     Pr>z
 ---------+-------------------------------------------------
       z2 |   1000     0.99958     0.277     -3.246  0.99941
```

Neither `sktest` nor `sfrancia` reject normality, but `swilk` does, and resoundingly. We began this experiment expecting both `sfrancia` and `swilk` to reject, evidence that we were then going to use to support still using D'Agostino or `sktest` on grouped data. We can say that `swilk` should not be used on tied data, but `sfrancia` surprised us, at least until we examined the insides of the test and realized that the ranks were adjusted for ties. What now disturbed us was `sfrancia`'s *lack* of sensitivity to ties. Could it be that, with tied data, `sfrancia` is *less* likely to detect departures from normality than it should?

Thus, we repeated our 10,000 samples of size 100 experiment for the normal and t(5) distributions, but this time rounding the data into units of halves:

True Distribution	Test	1%	5%	10%	True Distribution	1%	5%	10%
Aggr. Normal					Aggr. t(5)			
	sktest	.023	.055	.085		.537	.626	.680
	D'Agostino	.019	.057	.101		.444	.583	.661
	swilk	.371	.709	.857		.419	.622	.749
	sfrancia	.003	.016	.033		.386	.528	.602

Beginning with the aggregated normal results, we are bothered by the far-from-nominal rejection rates of `swilk` and `sfrancia`; `swilk` rejects too much (as we suspected) while `sfrancia` rejects too seldom (which we did not). `sfrancia`'s reject-too-seldom problem carries over to the aggregated t(5) distribution. Thus, while impressed with `sfrancia`'s performance on nonaggregated data, we recommend not using `sfrancia` with aggregated data.

Royston, in summarizing the results he reported in *sg3.1*, recommended that `sktest` be withdrawn and provided other, not unconvincing evidence to justify his recommendation. We now respond to his recommendation:

1. We will not withdraw `sktest` nor the D'Agostino variant, `sktestd`, as we feel these programs still have a use with aggregate data.
2. We are convinced by Royston's findings on the instability of `sktest`. If we had to choose just one of `sktest` or `sktestd`, we would choose D'Agostino's `sktestd`. `sktestd` has the further advantage that it is from a published source.
3. Based on our findings and those of Royston (especially his Figure 2b), we are convinced that the D'Agostino's `sktestd` needs an empirical correction in the spirit of `sktest`. The correction provided by us, however, needs improvement. *[Along these lines, please see the following insert, sg3.5—Ed.]*
4. We are convinced by Royston's arguments and our own findings that Shapiro–Francia (`sfrancia`) is the preferred test for nonaggregate data. In the next release of our manuals, we will include words to that effect.
5. We are not fond of the Shapiro–Wilk (`swilk`) test—see tables.

Postscript: We communicated our results to Royston, which lead to the insert following this one along with a quick attempt to "fix" `swilk` for aggregate data in the same way that `sfrancia` is "fixed," namely to substitute a rank calculation that accounts for ties. This would lead us to change the words we wrote above, but not the conclusions. The new `swilk`, which is the one on the distribution disk, now understates rather than overstates rejection probabilities with grouped data. For the grouped normal, the rejection rates were .0006, .0068, and .0301 for the nominal rejection probabilities of .001, .01, and .05 in the grouped normal. While these rejection rates are similar to those for `sfrancia`, the modified `swilk` did not do so well when it came to rejection rates from the grouped t(5). The observed rejection rates were .191, .264, and .315.

References

D'Agostino, R. B., A. Belanger and R. B. D'Agostino, Jr. 1990. A suggestion for using powerful and informative tests of normality. *The American Statistician* 44(4): 316–321.

Gould, W. 1991. sg3: Skewness and kurtosis tests of normality. *Stata Technical Bulletin* 1: 20–21.

Royston, J. P. 1991. sg3.1: Tests for departure from normality. *Stata Technical Bulletin* 2: 16–17.

——. 1991. sg3.2: Shapiro–Wilk and Shapiro–Francia tests. *Stata Technical Bulletin* 3: 19.

sg3.5 Comment on sg3.4 and an improved D'Agostino test

Patrick Royston, Royal Postgraduate Medical School, London, FAX (011)-44-81-740 3119

As I made clear previously, I certainly think that `sktest` should go. Its null distribution is nowhere near N(0,1) as my Figures 1a and 1b in *sg3.1* indicated and therefore will still be wrong for aggregated data. It is misleading to give its p values for the individual $\sqrt{b_1}$ and b_2 statistics as well as for the combined test, since all of them are obviously way out. Moreover, one cannot usefully compare powers of tests when the nominal rejection rates differ markedly. I think the D'Agostino `sktestd` is less sinful, though still unsatisfactory and, as Gould and Rogers say in their point 3, in need of an empirical correction. I have concocted such a correction, which I present below in the form of a Stata ado-file.

Before turning to the correction, I wish to note that the 'alternative' (non-normal) distributions Gould and Rogers use in their power tables are biased towards symmetric, long-tailed distributions. Of the seven they quote, only two (χ^2 with 5 and 10 df) are skew and one (the uniform) is short-tailed. In practice, I find that skew distributions are much more common in real (e.g., medical) data than are symmetric ones. A typical model is the lognormal, in which $\log y$ is normal (or $\log(y-k)$ is normal, which gives the 3-parameter lognormal). Short-tailed symmetric distributions (e.g., the uniform) seem particularly rare, though I agree that long-tailed symmetric distributions are a reasonable model for contaminated normal data.

I have recently been performing some power comparisons using as alternates (1) skew long-tail, (2) symmetric long-tailed, and (3) skew short-tailed distributions. While (as shown by Gould and Rogers) the Shapiro–Wilk W is indeed poor for class (2) alternatives and W' is good, W and W' are about equally good for class (1) and W is much better than W' for class (3). The Anderson–Darling A^2 seems to perform similarly to W' but is never quite as good (its power is about 0.05 to 0.1 less than that of W' when the latter has power about 0.9 for a 5% significance test, $n = 100$). I think class (1) is the most important in practice, but it is a good question as to whether real data fall more commonly into class (2) or (3). I hope to publish an expanded version of the power results, with more tests being compared.

In summary, I do not necessarily agree with Gould and Rogers' conclusion that W is inferior to W' for unaggregated data.

Regarding aggregated data, I was not surprised that W rejected aggregated normal data too often but, like Gould and Rogers, I was surprised by the findings for W'. (I have published an adjustment to W to allow for aggregation, but I did not look at W'.) For now, the D'Agostino test is likely to be robust against aggregation and that justifies some effort being put into getting its null distribution right.

I have developed such an empirical correction and also, in the process, extended its range of application down from $n = 20$ to $n = 8$. I cannot go below $n = 8$ because the normalization for $\sqrt{b_1}$ (skewness) is not defined. Actually, the normalization for b_2 (kurtosis) is poor on its own below $n = 50$, let alone $n = 20$, but my correction compensates for the problem when skewness and kurtosis are combined to produce an approximate $\chi^2(2)$ statistic. It works satisfactorily to about the 0.995 centile, which should be good enough for practical purposes. Figures 2a and 2b repeat my previous Figures 2a and 2b, but include the modified D'Agostino test.

The syntax of use of `sktestdc` is similar to that of `sktest` and `sktestd`:

`sktestdc` *varlist* [`if` *exp*] [`in` *range*] [`=`*exp*] [`, noadjust`]

For each variable in *varlist*, `sktestdc` presents a test for normality based on skewness and another based on kurtosis and then combines the two tests for an overall test statistic. `=`*exp*, if specified, specifies the weight. `sktestdc` is a variation on `sktestd` presented in *sg3* and is based on D'Agostino, et al., referenced below. An empirical adjustment to the overall chi-square statistic and its P value are made unless the `noadjust` option is specified.

```
. sktestdc weight mpg
                Skewness/Kurtosis tests for Normality
                                                  ------- joint -------
 Variable |  Pr(Skewness)   Pr(Kurtosis)  adj chi-sq(2)  Pr(Chi-sq)
----------+-----------------------------------------------------------
   weight |      0.575         0.018            5.73       0.0570
      mpg |      0.002         0.080           10.97       0.0041
. sktestdc weight mpg, noadjust
                Skewness/Kurtosis tests for Normality
                                        ------- joint -------
 Variable |  Pr(Skewness)   Pr(Kurtosis)      chi-sq(2)  Pr(Chi-sq)
----------+-----------------------------------------------------------
   weight |      0.575         0.018            5.92       0.0519
      mpg |      0.002         0.080           13.13       0.0014
```

Notice that the adjustment may reduce the joint chi-square statistic making it less significant than before (it has a higher P value).

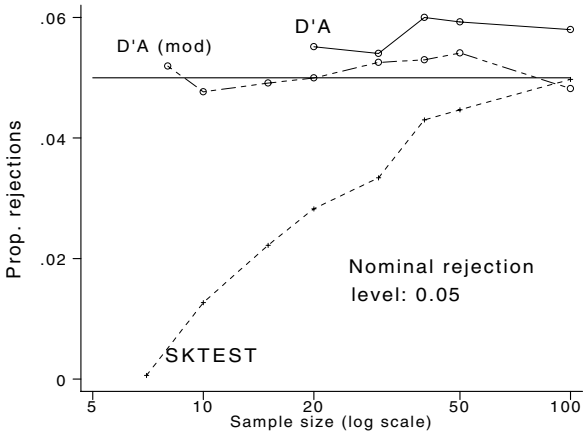


Figure 2a

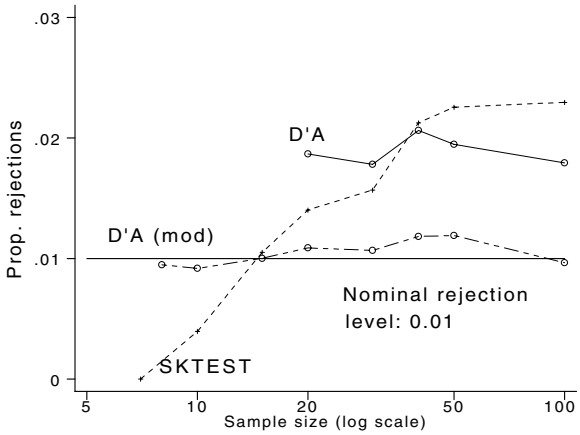


Figure 2b

References

D'Agostino, R. B., A. Belanger and R. B. D'Agostino, Jr. 1990. A suggestion for using powerful and informative tests of normality. *The American Statistician* 44(4): 316–321.

Gould, W. and W. Rogers. 1991. sg3.4: Summary of tests of normality. *Stata Technical Bulletin* 3: 20–23.

Royston, J. P. 1991. sg3.1: Tests for departure from normality. *Stata Technical Bulletin* 2: 16–17.

sg4	Confidence intervals for t-test

Richard Goldstein, Qualitas, Brighton, MA, EMAIL goldst@harvarda.bitnet

The syntax of `tci` is

`tci` *continuous_var group_var* [`if` *exp*] [`in` *range*]

There are many problems with any test of significance because not enough information is presented for adequate interpretation. There are at least two different ways that one can go in obtaining additional information: (1) analysis of the power of the test, or, better, the power function; (2) some estimate of the size of the effect. This latter can be done in several ways, but confidence intervals are the best known. This ado-file produces confidence intervals and graphs of the intervals for t tests.

`tci` gives 99.9%, 99%, 95%, 90%, 80%, 65% and 50% confidence intervals, and t test results, assuming that the variances are equal. A graph showing these limits is produced at the end; this is saved as `#.gph`, where # is replaced by $1 \ldots n$ for the number of continuous variables you run for this ado-file without leaving Stata. If you leave Stata and then re-enter and try to run this you will get an error message when the program tries to save the first graph unless you have changed the name of the graphs previously saved. The ado-file is written this way assuming you want to keep old graphs, or at least you do not want to automatically destroy them.

Your *group_var* will automatically and temporarily be replaced by a 0-1 version so that the differences shown in the output will be correct based on simple calculations. The 0-1 code will be 0 for the smallest of the first two codes found and the 1 will be for the largest. The code will even reformulate categorical variables with more than two categories. In this case the code automatically uses the first two categories, though using `if` or `in` you can choose other categories, as shown in the example below. If you want this handled some other way, then make your own dummy variable prior to calling `tci`. `tci` automatically restores your data when it completes processing from a temporary copy it made.

Note that the xlabels and the top labels are not complete since this would lead to overlap of .99 and .999—however there are tick marks at each of the confidence levels graphed; seven levels are graphed, the same as the numbers that are shown (see the examples below). Note also that the graphs have two horizontal guidelines: (1) the difference between the two means; (2) at zero difference.

`tci` calls another ado-file, `invt.ado`, provided to me by Bill Gould, that is a general purpose inverse-t statistical function. This file will automatically be installed when you install `tci`. *[The use of the custom* `invt.ado` *was not really necessary, the Stata* `emdef` *command will do the same thing; see* `help emdef`*—Ed.]*

Examples

The examples use the Stata-provided `auto.dta` data set. Note that the confidence intervals are shown first, followed by the two group means, and then the t test results.

```
. use auto
(1978 Automobile Data)
. tci mpg for

                                        Lower Limit     Upper Limit
The level(.999) confidence interval is:      0.2701          9.6215
The level(.99) confidence interval is:       1.3417          8.5499
The level(.95) confidence interval is:       2.2304          7.6612
The level(.90) confidence interval is:       2.6760          7.2156
The level(.80) confidence interval is:       3.1840          6.7077
The level(.65) confidence interval is:       3.6644          6.2272
The level(.50) confidence interval is:       4.0224          5.8692
mean for first group is   19.8269    mean for second group is   24.7727
The difference is:    4.9458   t:   3.6308  and p-value is 0.0005 (N=74)
(1978 Automobile Data)
```

The second example uses `if` to test two of five groups:

```
. tci mpg rep78 if rep78==3 | rep78==5
Following are confidence intervals for the difference:
                                        Lower Limit     Upper Limit
The level(.999) confidence interval is:       .7051        15.1555
The level(.99) confidence interval is:       2.5028        13.3578
The level(.95) confidence interval is:       3.8780        11.9826
The level(.90) confidence interval is:       4.5548        11.3058
The level(.80) confidence interval is:       5.3186        10.5420
The level(.65) confidence interval is:       6.0352         9.8254
The level(.50) confidence interval is:       6.5663         9.2943
mean for first group is   19.4333    mean for second group is   27.3636
The difference is:    7.9303   t:  3.9584  and p-value is 0.0004 (N=41)
(1978 Automobile Data)
```

The data set name is shown at the end since the ado file resets the data file to the way it was before running the `tci`; thus, all new variables disappear. However, the nonlocal macros, including the graph counter, do not disappear; they can be examined by typing `mac list` at the Stata command prompt.

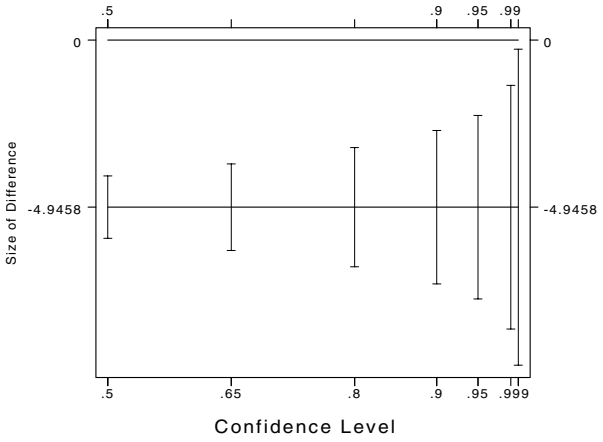


Figure 1

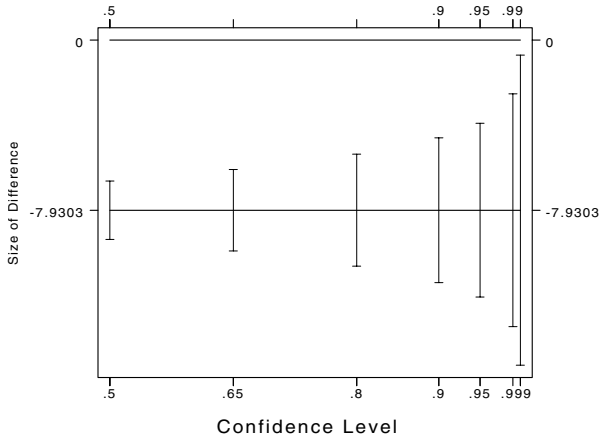


Figure 2

snp2 — Friedman's ANOVA test & Kendall's coefficient of concordance

Richard Goldstein, Qualitas, Brighton, MA, EMAIL goldst@harvarda.bitnet

The syntax of `friedman` is

`friedman` *varlist* [`in` *range*] [`if` *exp*]

`friedman` estimates Friedman's nonparametric two-way analysis of variance and Kendall's Coefficient of Concordance (a descriptive measure of the agreement between k sets of rankings). The two tests are equivalent and one p-value is given for both. Note that the value of Kendall's statistic must be between 0 and 1 and therefore may be easier to interpret.

Missing values are not allowed. I have tried to trap this, but may not find all cases—if you ever obtain a negative result for the test statistics, then you have at least one missing value.

For Friedman's test, the variables indicate treatments while the cases indicate blocks, subjects or samples. For Kendall's coefficient, the variables are the rankers or judges or tests, while the cases are the things being judged or ranked. For example, one common use is to compare the rankings of students on each of several tests; the following is taken from J. D. Gibbons (1985), *Nonparametric Statistical Inference*, 2nd edition, NY: Marcel Dekker, Inc., pp. 326–327.

Example

Consider the following data:

	Student							
TEST	1	2	3	4	5	6	7	8
1	90	60	45	48	58	72	25	85
2	62	81	92	76	70	75	95	72
3	60	91	85	81	90	76	93	80

If you enter this as three variables, then you can just type "`friedman it varlist`" and all is fine. If you enter the data as 8 variables, one per student, then use the "`xpose, c`" command first. You can then enter the `friedman` command. Note that if you need to transpose the data, you should first eliminate all variables that you will not be analyzing. Otherwise you may get a surprise.

The above data set is included on the STB disk as `gibbons2.dta` as 8 variables. I also include Gibbons' example Friedman data set from p. 325 of her book as `gibbons1.dta`. This one also needs to be transposed prior to testing. This code is written expecting the raters to be the variables. This is because the `genrank` program ranks variables across cases, not the other way around. Note that in other programs you would do things differently–thus, in Systat enter this data set as 8 variables to obtain the same results. The correct results for the above data are

$$\text{Friedman} = 2.8889$$
$$\text{Kendall} = 0.1367$$
$$\text{p-value} = 0.8951$$

Kendall's coefficient of concordance ranges from 0 to 1, with 0 meaning no agreement across raters (judges). The null hypothesis (Friedman) is that the treatments are equal; the null hypothesis (Kendall) is that there is no agreement between rankings or test results.

I do not produce the sums of ranks for each case, though other software does. To add it to your output, add the line `li sum_rs` near the end of the `friedman.ado` file. I recommend either just before or just after the current display lines.

The p-value is an approximation, though it appears to be pretty good as long as there are at least 8 cases and 3 variables.

This code calls a `crc` ado-file called `genrank` and an ado-file called `genvsum` which is 'stolen' from the `crc` ado-file called `genvmean`. `genvsum.ado` is included, as are two data files: `gibbons2.dta` (above) and `gibbons1.dta` which is from the same book and is her other example. Also included is a help file called `kendall.hlp` which just sends you to the full `friedman.hlp` file. Finally, `lehmann1.dta` and `noether1.dta` are the sample files from those books (citations below).

The code was tested against the Friedman/Kendall results in Systat 5.0, SPSS 4.0, NCSS 5.3 and BMDP/90. The code here agrees with the results from Systat in every case for both tests; NCSS only offers the Friedman test, but that also agrees in every case tried. The results from BMDP agree in every case except where at least one column has a standard deviation of 0.0—in this case, BMDP drops the case(s), though no one else does. Certainly if one is primarily interested in the Kendall interpretation, i.e., agreement, then dropping cases on which all judges agree makes no sense. There were relatively minor differences between all other codes and SPSS; the latter using a correction for ties that is not found in other packages.

Different formulae appear in different texts, though several texts use one particular different formula, including E. Lehmann (1975), *Nonparametrics*, Oakland: Holden–Day, p. 263; G. Noether (1976), *Introduction to Statistics: A Nonparametric Approach*, 2nd ed., Boston: Houghton Mifflin Co., p. 182; and F. Mosteller, F and R. E. K. Rourke (1973), *Sturdy Statistics*, Reading: Addison–Wesley, p. 229. The code here matches the examples in Lehmann and Noether; there is no worked example in Mosteller & Rourke. A different formula and quite a different presentation is in W. J. Conover, (1980), *Practical Nonparametric Statistics*, 2nd ed., NY: John Wiley & Sons, p. 299. Conover gives a formula that allegedly gives the relationship between his formula and the others, but I still cannot match his example on p. 301.

snp3	Phi coefficient (fourfold correlation)

Richard Goldstein, Qualitas, Brighton, MA, EMAIL goldst@harvarda.bitnet

The syntax of `phi` is

`phi` *categorical_var categorical_var* [`if` *exp*] [`in` *range*] [`,` *options*]

You can enter any `tabulate` option except `chi`; the `chi` option is built into the command since the statistic is needed to calculate phi. You can use the `all` option, however. (See `help tabulate`.)

`phi` calculates ϕ, or correlation coefficient, for a 2×2 table, along with the χ^2 statistic. This is also Cohen's effect size, w, for those using J. Cohen (1988), *Statistical Power Analysis for the Behavioral Sciences*, 2nd edition, Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers. You might also need this if you are using software to calculate power that requires that you know w.

The ϕ coefficient is equal to $\sqrt{\chi^2/N}$. If you have a 2×2 table, this is equal to the correlation coefficient for these variables (see the example below). Since this is a correlation coefficient, its square is meaningful and is provided in the output. If either the rows or the columns are greater than 2, then this program supplies Cramer's ϕ' as well as Cohen's w. Note that if either rows or columns is greater than 2, then w does not equal ϕ' and ϕ' is not the correlation coefficient either; also, its square is not meaningful and is not provided.

Examples

A 2×2 table, followed by the correlation:

```
. phi female ra if staff==0
           | Returned Admin.
    female |        no        yes |     Total
-----------+----------------------+----------
      Male |        27         16 |        43
    Female |        31          3 |        34
-----------+----------------------+----------
     Total |        58         19 |        77
          Pearson chi2(1) =   8.2311   Pr = 0.004
phi = Cohen´s w = fourfold point correlation = 0.3270   phi-squared = 0.1069
. corr female ra if staff==0
(obs=77)
        |   female       ra
--------+------------------
 female |   1.0000
     ra |  -0.3270   1.0000
```

A 2×4 table followed by its correlation:

```
. phi female srank if staff==0
           | srank
    female |         0          1          2          3 |     Total
-----------+--------------------------------------------+----------
      Male |        17         14          9          3 |        43
    Female |        17         17          0          0 |        34
-----------+--------------------------------------------+----------
     Total |        34         31          9          3 |        77
          Pearson chi2(3) =  11.3940   Pr = 0.010
Cramer´s phi-prime =  0.3847     Cohen´s w = 0.3847
. corr female srank if staff==0
(obs=77)
        |   female    srank
--------+------------------
 female |   1.0000
  srank |  -0.2786   1.0000
```

sqv1.2	Additional logit regression diagnostic - Cook's Distance

Joseph Hilbe, Editor, STB, FAX 602-860-1446

I have added Cook's Distance to the list of diagnostic variables created by the `logiodd2` command. The latter command calls `extlogit`, which calculates a variety of logistic diagnostics. A revised `extlogit.ado` has been placed on the STB-3 diskette. Refer to *sqv1* in STB-1 for proper use and notice as to how the program relates duplicate covariate patterns to influence statistics.