STATA TECHNICAL BULLETIN

November 2000

STB-58

A publication to promote communication among Stata users



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

| dm65.1 | Update to a program for saving a model fit as a dataset |
|---|---|

Roger Newson, Guy's, King's and St Thomas' School of Medicine, London, UK, roger.newson@kcl.ac.uk

**Abstract:** The `parmest` command introduced in Newson (1999) has been updated for Stata 6.0.

**Keywords:** confidence intervals, estimation results.

Introduced in Newson (1999), `parmest` saves the most recent estimation results to a dataset with one observation per parameter and is typically used (together with `graph`) to produce confidence interval plots. It has been updated for Stata 6.0. Two bugs have been corrected. First, `parmest` now works on estimation results from multi-equation models (such as those fitted by `mlogit`), which previously caused it to fail. Second, when the results are saved using the `saving()` option, `parmest` no longer saves any temporary variables with confusing names. To rule out these bugs and many others, `parmest` has been tested extensively using the Stata 6.0 certification script utility (see the on-line `help cscript`).

## Saved results

`parmest` now saves in `r()`:

Scalars

| | |
|---|---|
| `r(dof)` | Degrees of freedom for $t$ distribution used for confidence intervals (0 if normal distribution used) |
| `r(nparm)` | Number of parameters estimated |
| `r(level)` | Value of `level` option (confidence level for CIs) |

Macros

| | |
|---|---|
| `r(eform)` | Value of `eform` option (`eform` if set, otherwise empty) |

## Acknowledgment

I would like to thank Vince Wiggins of Stata Corporation for his helpful advice about handling equation names containing spaces (which are often produced by `mlogit` for outcomes with value labels).

## References

Newson, R. 1999. dm65: A program for saving a model fit as a dataset. *Stata Technical Bulletin* 49: 2–5. Reprinted in *Stata Technical Bulletin Reprints*, vol. 9, pp. 19–23.

| dm82 | Simulating two- and three-generation families |
|---|---|

Jisheng Cui, University of Melbourne, Australia, j.cui@gpph.unimelb.edu.au

**Abstract:** The commands `simuped2` and `simuped3` for simulating two- and three-generation families are introduced and illustrated.

**Keywords:** family data, generations, simulation.

## Introduction

Simulation of family data (pedigree) is sometimes required in genetic epidemiology research. Generation of family data using Stata has advantages over using other computer languages because of the random number generators of probability distributions built into Stata. Here we present two programs, `simuped2` and `simuped3`. The former is used to simulate two-generation families, and the latter is used to simulate three-generation families. Figures 1 and 2 give schematic illustrations of the pedigree structure of the families generated by these programs. A circle represents a female, while a square represents a male. There is a marriage in the second generation in Figure 2.

(*Continued on next page*)

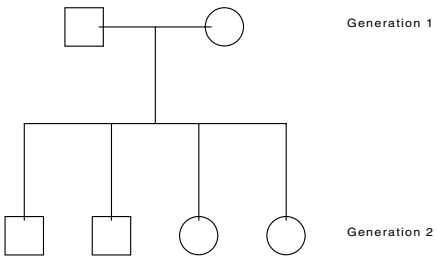



Figure 1. Schematic illustration of a two-generation family.

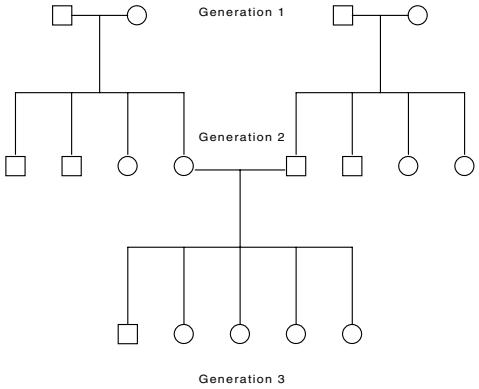



Figure 2. Schematic illustration of a three-generation family.

## Syntax

`simuped2` $\#_{\text{Age1}}$ $\#_{\text{Std1}}$ $\#_{\text{Age2}}$ $\#_{\text{Std2}}$ [`, reps(#) saving(`*filename*`) alle(#) sib(#)` ]

`simuped3` $\#_{\text{Age1}}$ $\#_{\text{Std1}}$ $\#_{\text{Age2}}$ $\#_{\text{Std2}}$ $\#_{\text{Age3}}$ $\#_{\text{Std3}}$ [`, reps(#) saving(`*filename*`) alle(#) sib(#) si3(#)` ]

## Description

`simuped2` and `simuped3` are immediate commands used for generating two- and three-generation family data, respectively. For each person in a family, the sex is generated by a probability of 0.5. The age of a person is generated according to a normal distribution, with means $\#_{\text{Age1}}$, $\#_{\text{Age2}}$, and $\#_{\text{Age3}}$ for the first, second, and third generations. The standard deviations of the ages are given by $\#_{\text{Std1}}$, $\#_{\text{Std2}}$, and $\#_{\text{Std3}}$, respectively.

The number of siblings in a generation is a random number, distributed according to a Poisson distribution. The mean sizes of the siblings in the second- and third-generation are given by `sib(#)` and `si3(#)`, respectively.

Hardy–Weinberg equilibrium is assumed for the genotypic distribution of people in the first generation (see, for example, Elandt-Johnson 1971). The allele frequency of a biallelic locus A is given by the argument `alle(#)`, denoted as $p$. The frequencies of genotypes AA, Aa and aa in the first generation are given by $p^2$, $2p(1-p)$ and $(1-p)^2$, respectively. The genotype of a person in the second- and third-generation is generated according to the Mendelian inheritance, that is, a person inherits the allele A from the father (or mother) with probability 0.5.

The simulated family data are saved in a file specified by `saving(`*filename*`)`, and the number of replications is specified by `reps(#)`.

## Options

`reps(#)` specifies the number of simulated families. The default value is 100.

`saving(`*filename*`)` specifies the file into which the simulated data are saved. The default file name is `temp.dta`.

`alle(#)` specifies the allele frequency of a biallelic locus A. The default value is 0.1.

`sib(#)` specifies the mean number of siblings in the second generation. The default value is 3.

`si3(#)` specifies the mean number of siblings in the third generation. It is only used in `simuped3`. The default value is 3.

## Remarks

`simuped2` and `simuped3` simulate the basic quantities of a person, such as age, sex, and genotype, which are useful in genetic epidemiology research. Further quantities of interest, such as the disease status of a person, can be simulated based on these basic family data. However, it sometimes requires specific models for the affect of a disease and the model of the natural mortality of a person. We do not include the disease status in the programs because that would not make `simuped2` and `simuped3` of as general use.

## Example 1

We simulate 1000 two-generation families. The mean age and standard deviation of people in the first- and second-generation are 70, 10 and 40, 10, respectively. The frequency of an allele A is assumed to be 0.05. The mean number of siblings in the second generation is 5. The simulated family data are saved into the file `output.dta`.

```
. simuped2 70 10 40 10, reps(1000) sav(output) alle(0.05) sib(5)
. use output
. describe
Contains data from output.dta
  obs:         6,818
 vars:             6                          23 Oct 2000 07:51
 size:       177,268 (83.0% of memory free)
-------------------------------------------------------------------------------
   1. famid     float  %9.0g
   2. id        float  %9.0g
   3. degree    float  %9.0g
   4. female    float  %9.0g
   5. age       float  %9.0g
   6. genotype  str2   %9s
-------------------------------------------------------------------------------
Sorted by:
. list
         famid         id     degree     female        age   genotype
   1.        1          1          1          0         73         aa
   2.        1          2          1          1         75         aa
   3.        1          3          2          0         28         aa
   4.        1          4          2          1         43         aa
   5.        1          5          2          1         25         aa
   6.        2          1          1          0         64         aa
   7.        2          2          1          1         58         aa
   8.        2          3          2          0         38         aa
   9.        2          4          2          1         46         aa
  10.        2          5          2          0         51         aa
  (output omitted)
6809.      999          2          1          1         51         aa
6810.      999          3          2          0         50         aa
6811.      999          4          2          1         38         aa
6812.      999          5          2          0         37         aa
6813.      999          6          2          1         41         aa
6814.     1000          1          1          0         74         aa
6815.     1000          2          1          1         70         aa
6816.     1000          3          2          1         38         aa
6817.     1000          4          2          1         41         aa
6818.     1000          5          2          1         38         aa
```

A total of 6,818 individuals are generated in the 1,000 families. The variable `famid` represents the family identification of the simulated family, while `id` represents the personal identification within each family, `degree` represents the generation a person belongs to, `female` is one or zero depending on whether or not a person is a female, `age` represents the simulated age, and `genotype` represents a person's genotype.

## Example 2

We simulate 2,000 three-generation families. The mean age of people in the first-, second- and third-generation are 80, 50 and 20, respectively. Their standard deviation is assumed to be 10 across all generations. The frequency of an allele A is assumed to be 0.1. The mean number of siblings in the second- and third-generation are 4 and 3.5, respectively.

```
. set memory 50m
. simuped3 80 10 50 10 20 10, reps(2000) alle(0.1) sib(4) si3(3.5)
. use temp
. describe
Contains data from temp.dta
  obs:        29,667
 vars:             8                          23 Oct 2000 18:38
 size:     1,008,678 (98.1% of memory free)
-------------------------------------------------------------------------------
   1. famid     float  %9.0g
   2. id        float  %9.0g
   3. degree    float  %9.0g
```

```
   4. family    float  %9.0g
   5. female    float  %9.0g
   6. marry     float  %9.0g
   7. age       float  %9.0g
   8. genotype  str2   %9s
-------------------------------------------------------------------------------
Sorted by:

. list
          famid         id     degree     family     female      marry        age   genotype
    1.        1          1          1          1          0          1         85         aa
    2.        1          2          1          1          1          1         68         aa
    3.        1          3          1          2          0          1         83         Aa
    4.        1          4          1          2          1          1         65         Aa
    5.        1          5          2          1          1          0         51         aa
    6.        1          6          2          1          0          0         61         aa
    7.        1          7          2          1          1          0         43         aa
    8.        1          8          2          1          1          0         33         aa
    9.        1          9          2          1          1          1          4         aa
   10.        1         10          2          1          1          0         25         aa
  (output omitted)
29658.     2000         10          2          2          0          1         53         aa
29659.     2000         11          2          2          1          0         64         aa
29660.     2000         12          2          2          1          0         32         aa
29661.     2000         13          2          2          0          0         39         aa
29662.     2000         14          3          0          0          0         30         aa
29663.     2000         15          3          0          0          0         12         aa
29664.     2000         16          3          0          0          0         34         aa
29665.     2000         17          3          0          0          0          5         aa
29666.     2000         18          3          0          1          0         27         aa
29667.     2000         19          3          0          0          0         11         aa
```

In this example, we do not specify the output file, so the simulated family data are saved into `temp.dta`. The extra variable named `marry` is produced by `simuped3` compared with `simuped2`. It indicates the marriage status of a person.

## References

Elandt-Johnson, R. 1971. *Probability Models and Statistical Methods in Genetics.* New York: John Wiley & Sons.

| gr45 | A turnip graph engine |
|---|---|

Steven Woloshin, VA Outcomes Group, VA Medical Center, White River Junction, VT, steven.woloshin@dartmouth.edu

**Abstract:** A new graphical description called a turnip graph for studying the distribution of a variable is introduced and illustrated.

**Keywords:** descriptive statistics, histogram, stem and leaf plot, turnip plot.

## Syntax

`turnip` *varname* [`if` *exp*] [`, resolution(#) truev(#)` *graph_options* ]

## Description

`turnip` creates a turnip-style graph for the variable *varname*. The range of the variable is divided into intervals which are placed on the vertical axis of the plot. Symbols are plotted horizontally next to each interval reflecting the number of observations in that interval. Thus one can use this plot in conjunction with histograms, boxplots, stem-and-leaf plots, and so on, to study the distribution of a variable.

## Options

`resolution(#)` specifies the resolution of the graph, that is, the width of the intervals to be used. The default value is $0.4s$, where $s$ is the standard deviation of the variable. Since `resolution` rounds the data, the graph in essence displays the frequency of observations falling within each resolution unit. The user can avoid any such rounding (that is, display the frequency of each value in the data) by specifying the resolution width as any negative number.

`truev(#)` specifies a value that can be used to divide a variable into three parts: one exactly equal to `truev`, one that is greater than `truev`, and one that is less than `truev`. For example, suppose you are trying to display change scores and want to show which observations are above or below zero. Using `truev(0)` ensures that only zero values are graphed at zero; other values which would round to zero are set at the next appropriate category of *varname*.

*graph_options* are any of the standard graphics options except for `by`. In addition to the usual function of `yline`, `turnip` allows users to specify special values; specifically, `mean` or `median`. If either of these is specified, the corresponding value is added to `ylabel` so the value is displayed on the $y$-axis. Note that the default $x$-axis label ranges between plus and minus twice the maximum number of observations in an interval, while the default $y$-axis label adds 25% of the range above and below the maximum and minimum values of the variable.

## Examples

Issuing the command

```
. turnip mpg
```

gives the output for the `mpg` variable in Stata's auto data

```
# of observations per turnip row is the frequency below:
        mpg |      Freq.     Percent        Cum.
------------+-----------------------------------
   11.57101 |          2        2.70        2.70
   13.88521 |          8       10.81       13.51
   16.19941 |          8       10.81       24.32
   18.51361 |         17       22.97       47.30
   20.82781 |          8       10.81       58.11
   23.14201 |         12       16.22       74.32
   25.45621 |          8       10.81       85.14
   27.77042 |          3        4.05       89.19
   30.08462 |          4        5.41       94.59
   34.71302 |          3        4.05       98.65
   41.65562 |          1        1.35      100.00
------------+-----------------------------------
      Total |         74      100.00
options:  xlabel(-16  16) ylabel(9 45 )
resolution:  2.314201283894056
```

and the graph in Figure 1; see below.

We can produce a similar graph with greater resolution and with a horizontal line at the median of `mpg` by

```
. turnip mpg, res(1.5) yline(median)
# of observations per turnip row is the frequency below:
        mpg |      Freq.     Percent        Cum.
------------+-----------------------------------
         12 |          2        2.70        2.70
       13.5 |          6        8.11       10.81
         15 |          2        2.70       13.51
       16.5 |          8       10.81       24.32
         18 |          9       12.16       36.49
       19.5 |         11       14.86       51.35
         21 |          5        6.76       58.11
       22.5 |          8       10.81       68.92
         24 |          4        5.41       74.32
       25.5 |          8       10.81       85.14
       28.5 |          4        5.41       90.54
         30 |          2        2.70       93.24
       31.5 |          1        1.35       94.59
       34.5 |          3        4.05       98.65
       40.5 |          1        1.35      100.00
------------+-----------------------------------
      Total |         74      100.00
options:  yline(20) xlabel(-10  10) ylabel( 20 9 43 )
resolution:  1.5
yline:  20  (median)
```

which gives the graph in Figure 2.

(*Continued on next page*)

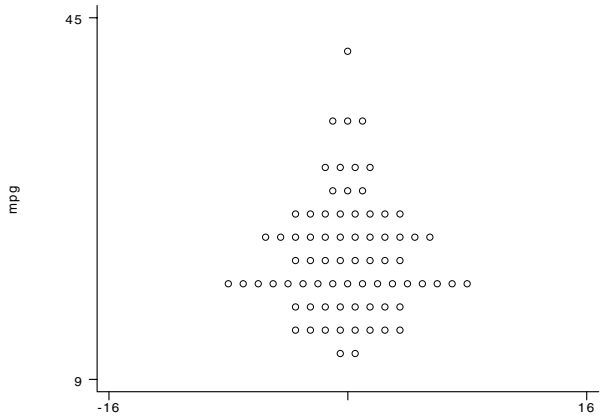



Figure 1. A simple turnip plot for mpg for the auto data.

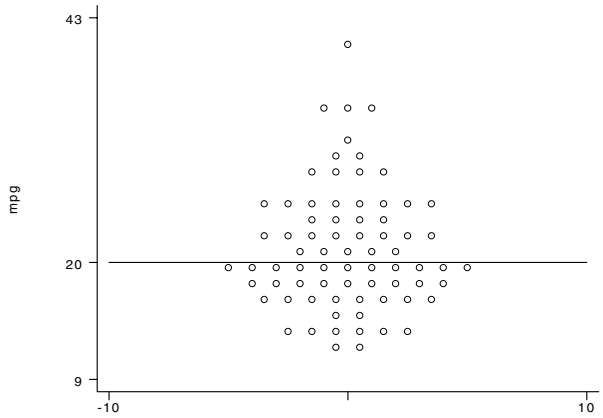



Figure 2. Higher resolution graph for mpg with line at median.

Finally, we do a turnip plot for only foreign cars with a specified $x$-axis label and a horizontal line at the mean.

```
. turnip mpg if foreign==1, xlabel(-10 10) yline(mean)
# of observations per turnip row is the frequency below:
```

| mpg | Freq. | Percent | Cum. |
|---|---|---|---|
| 13.88521 | 1 | 4.55 | 4.55 |
| 16.19941 | 2 | 9.09 | 13.64 |
| 18.51361 | 2 | 9.09 | 22.73 |
| 20.82781 | 2 | 9.09 | 31.82 |
| 23.14201 | 4 | 18.18 | 50.00 |
| 25.45621 | 5 | 22.73 | 72.73 |
| 27.77042 | 1 | 4.55 | 77.27 |
| 30.08462 | 2 | 9.09 | 86.36 |
| 34.71302 | 2 | 9.09 | 95.45 |
| 41.65562 | 1 | 4.55 | 100.00 |
| Total | 22 | 100.00 | |

```
options:  xlabel(-10 10) yline(24.7) ylabel( 24.77272727272727 11 44 )
resolution:  2.314201283894056
yline:  24.77272727272727  (mean)
```

This gives the plot in Figure 3.

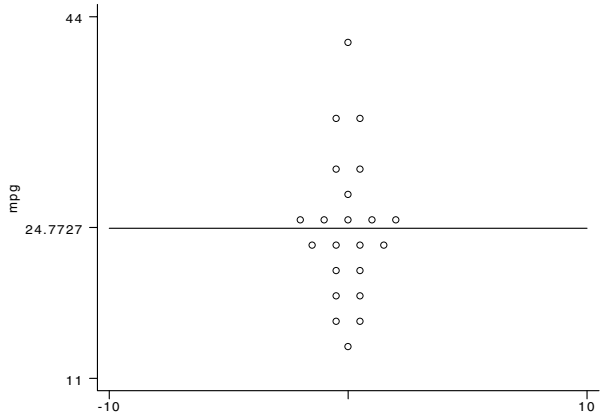



Figure 3. Turnip plot for foreign cars with specified $x$-axis label and a horizontal line at the mean.

## Acknowledgment

Turnip graphs are the graph of choice in the Dartmouth Atlas of Healthcare (American Hospital Publishing Inc., Chicago, IL 1996). The term turnip graph was coined by Jack Wennberg and colleagues at the Center for the Evaluative Clinical Sciences, Dartmouth Medical School, because the display sometimes reminded them of turnips. Other suggested names included carrots, flying saucers, and the Stealth Bomber.

## sbe19.3 Tests for publication bias in meta-analysis: erratum

Thomas J. Steichen, RJRT, steicht@rjrt.com

**Abstract:** This insert provides a correction to the help file for the `metabias` command introduced in Steichen (1998) and modified in Steichen et al. (1998) and Steichen (2000).

**Keywords:** meta-analysis, publication bias, Egger, Begg.

### Description

As one form of data input, `metabias` allows the user to provide effect estimates, *theta*, and standard errors, *se_theta*. The published help file included a note stating that data in binary count format could be converted to the effect format used in `metabias` by use of program `metan` (Bradburn et al. 1998). This note stated that `metan` automatically adds variables for *theta* and *se_theta* to the raw dataset, naming them `_ES` and `_seES`, and that these variables could be provided to `metabias` using its default input method.

This is not correct. When processing binary data, `metan` automatically adds variables for *exp(theta)* and *se_theta*, that is, it is *exp(theta)* that is stored in variable `_ES`, not *theta*. The user must manually transform these exponentiated values back to *theta* format using Stata's `log()` function (or, equivalently, `ln()` function) before providing them to `metabias`. This additional step is now documented in the help file.

### Acknowledgment

I am grateful to Dr. John Moran for indirectly alerting me to this error.

### References

Bradburn, M. J., J. J. Deeks, and D. G. Altman. 1998. sbe24: metan—an alternative meta-analysis command. *Stata Technical Bulletin* 44: 4–15. Reprinted in *Stata Technical Bulletin Reprints*, vol. 8, pp. 86–100.

Steichen, T. J. 1998. sbe19: Tests for publication bias in meta-analysis. *Stata Technical Bulletin* 41: 9–15. Reprinted in *Stata Technical Bulletin Reprints*, vol. 7, pp. 125–133.

——. 2000. sbe19.2: Updates of tests for publication bias in meta-analysis. *Stata Technical Bulletin* 57: 4.

Steichen, T. J., M. Egger, and J. Sterne. 1998. sbe19.1: Tests for publication bias in meta-analysis. *Stata Technical Bulletin* 44: 3–4. Reprinted in *Stata Technical Bulletin Reprints*, vol. 8, pp. 84–85.

## sbe39.1 Nonparametric trim and fill analysis of publication bias in meta-analysis: erratum

Thomas J. Steichen, RJRT, steicht@rjrt.com

**Abstract:** This insert provides a correction to `metatrim` (Steichen 2000) and to its help file. `metatrim` implements the Duval and Tweedie (2000) nonparametric "trim and fill" method of accounting for publication bias in meta-analysis.

**Keywords:** meta-analysis, publication bias, nonparametric, data augmentation.

### Description

As one form of data input, `metatrim` allows the user to provide effect estimates, *theta*, and standard errors, *se_theta*. Both Steichen (2000) and its accompanying help file included a note stating that data in binary count format could be converted to the effect format used in `metatrim` by use of program `metan` (Bradburn et al. 1998). This note stated that `metan` automatically adds variables for *theta* and *se_theta* to the raw dataset, naming them `_ES` and `_seES`, and that these variables could be provided to `metatrim` using its default input method.

This is not correct. When processing binary data, `metan` automatically adds variables for *exp(theta)* and *se_theta*. That is, it is *exp(theta)* that is stored in variable `_ES`, not *theta*. The user must manually transform these exponentiated values back to *theta* format using Stata's `log()` function (or, equivalently, `ln()` function) before providing them to `metatrim`. This additional step is now documented in the help file.

## Acknowledgment

I am grateful to Dr. John Moran for alerting me to this error.

## References

Bradburn, M. J., J. J. Deeks, and D. G. Altman. 1998. sbe24: metan—an alternative meta-analysis command. *Stata Technical Bulletin* 44: 4–15. Reprinted in *Stata Technical Bulletin Reprints*, vol. 8, pp. 86–100.

Duval, S. and R. Tweedie. 2000. A nonparametric "trim and fill" method of accounting for publication bias in meta-analysis. *Journal of the American Statistical Association* 95(449): 89–98.

Steichen, T. J. 2000. sbe39: Nonparametric "trim and fill" analysis of publication bias in meta-analysis. *Stata Technical Bulletin* 57: 8–14.

| sg84.3 | Concordance correlation coefficient: minor corrections |
|---|---|

Thomas J. Steichen, RJRT, steicht@rjrt.com
Nicholas J. Cox, University of Durham, UK, n.j.cox@durham.ac.uk

**Abstract:** This insert fixes some bugs and corrects some defaults affecting detailed user control of graphical output in the program for concordance correlation. The numerical calculations were not changed.

**Keywords:** concordance correlation, graphics, measurement comparison.

## Description

`concord` computes the concordance correlation coefficient for agreement on a continuous measure obtained by two persons or methods and provides optional graphical displays. A full description of the method and of the operation of the command was given by Steichen and Cox (1998a), with revisions and updates in Steichen and Cox (1998b, 2000).

This insert fixes some small bugs affecting detailed user control of graphical display through the `connect()`, `symbol()`, and `pen()` options. It also corrects a few of the corresponding default values.

Specifically, the default *graph_options* for `graph(loa)` now include `connect(lll.l)`, `symbol(iiioi)`, and `pen(35324)` for the lower confidence interval limit, mean difference, upper confidence interval limit, data points, and regression line (if requested) respectively, along with default titles and labels. Further, the visual characteristics of the data points and line in the normal probability plot now follow those of the data points and regression line in the associated loa (limits-of-agreement) plot.

## Acknowledgment

We are grateful to Sónia Dória Nóbrega for alerting us to the bugs.

## References

Steichen, T. J. and N. J. Cox. 1998a. sg84: Concordance correlation coefficient. *Stata Technical Bulletin* 43: 35–39. Reprinted in *Stata Technical Bulletin Reprints*, vol. 8, pp. 137–143.

——. 1998b. sg84.1: Concordance correlation coefficient, revisited. *Stata Technical Bulletin* 45: 21–23. Reprinted in *Stata Technical Bulletin Reprints*, vol. 8, pp. 143–145.

——. 2000. sg84.2: Concordance correlation coefficient: update for Stata 6. *Stata Technical Bulletin* 54: 25–26. Reprinted in *Stata Technical Bulletin Reprints*, vol. 9, pp. 169–170.

| sg97.2 | Update to formatting regression output |
|---|---|

John Luke Gallup, Harvard University, jgallup@hiid.harvard.edu

**Abstract:** An update to the `outreg` command is described.

**Keywords:** regression output.

I have updated `outreg`, a program described in Gallup (1998, 1999) that writes regression output to a text file. New features allow including user-specified statistics and notes, 10% asterisks, table and column titles, scientific notation for coefficient estimates, and reporting of confidence interval and marginal effects. I have also fixed bugs in the reporting results from `dprobit`, `heckman`, and `heckprob` regressions, and `outreg` can now be used after `dlogit2`, `dprobit2`, and `dmlogit2`. Because `outreg` has been so extensively modified, we describe its full syntax.

## Syntax

outreg [*varlist*] using *filename* [, nolabel title(*textlist*) ctitle(*textlist*) nonotes addnote(*textlist*)

bdec(*numlist*) bfmt(*textlist*) coefastr {se | pvalue} ci tdec(#) noparen bracket

{noaster | 3aster | 10pct} sigsymb(*textlist*) nocons nonobs noni nor2 adjr2 rdec(#)

addstat(*text*, # [, *text*, #, ...]) adec(#) eform marginal onecol xstats

comma append replace ]

outreg is available after any estimation command. A *textlist* is a list of text separated by commas. It is similar to a *numlist* or a *varlist*, but commas are required. Each text element in the list does not need to be enclosed in quotation marks unless the text contains commas or parentheses.

## Description

outreg formats regression output as it is presented in most documents: $t$ statistics or standard errors in parentheses under each coefficient, asterisks indicating coefficients statistically different from zero, and summary statistics like $R$-squared at the bottom. The formatted output is written to a tab- or comma-separated ASCII file, which can then be loaded into word processing or spreadsheet programs to be converted directly to a table. For example, the output file can be opened in Microsoft Word, the text selected, and then converted to a table by choosing "Table", then "Convert Text to Table". Note that when loading the output into a spreadsheet, the parentheses around the $t$ statistics may convert to negative numbers.

outreg should work after any Stata estimation command. Like predict, outreg makes use of internally saved estimation results, so it should be invoked immediately after the estimation. In addition to coefficient estimates, by default outreg will report $t$ statistics with asterisks for standard significance levels (1% and 5%), numbers of observations, true $R$-squareds (no pseudo $R$-squareds), and the number of groups in panel estimation. The user can add their own chosen titles (*title* and *ctitle*), statistics (addstat), and notes (addnote) to the table, and change many other aspects of the output.

outreg rewards the use of variable labels (and value labels for mlogit, svymlog, and dlogit2). The variable labels are used in the output table (unless nolabel is chosen), providing more intelligible variable descriptions than 8-letter names. If different variables are assigned the same variable label (not usually done intentionally), and more than one regression is appended together, the coefficients and $t$ statistics will not be properly ordered. The solution is to use distinct variable labels or the nolabel option.

If *filename* is specified without an extension, .out is assumed.

Several regressions with differing variables can be combined into a single table with the append option.

If a *varlist* is specified, only the regression coefficients corresponding to the variables in *varlist* will be included in the table. The intercept coefficient is included as well unless the nocons option is chosen. This is probably most useful for excluding numerous dummy variable coefficients. Time series prefixes are not allowed when using an explicit *varlist*.

## Text-related options

nolabel specifies that variable names rather than variable labels be used to identify coefficients. It also suppresses the value labels of the dependent variable in mlogit and svymlog.

title(*textlist*) specifies a title or titles at the top of the regression table. The maximum title length is 80 characters. Additional characters will be cut off. Longer titles can be put in two or more title lines. When regression results are appended together, the table title(s) must be specified in the first outreg call; titles specified in subsequent outreg calls using the append option will be ignored. Note that when converting the outreg text output to a table in a word processor or a spreadsheet, it is easier to leave the title row out of the text selected for conversion.

ctitle(*textlist*) specifies the regression title above the coefficient column. By default if no column title is specified, the label or name of the dependent variable is displayed. Multiple column titles are only appropriate for multi-equation regressions, using one title per equation, and then only if not onecol.

nonotes specifies that notes explaining the $t$ statistics (or standard errors) and asterisks not be included.

addnote(*textlist*) specifies user-added notes to be displayed in new lines at the bottom of the outreg table. When regression results are appended together, addnote must be specified in the first outreg call; addnotes specified in subsequent outreg calls using the append option will be ignored. addnote is consistent with nonotes. A blank line can be inserted by including a blank within quotes as a note.

One technical note is in order. Text which includes quotation marks within the text (by means of double quotation) in `title`, `ctitle`, and `addnote` displays correctly in single regression tables but does not display correctly when subsequent regressions are appended using the `append` option.

### Coefficient options

`bdec(`*numlist*`)` specifies the number of decimal places reported for coefficient estimates. It also specifies the decimal places reported for standard errors or confidence intervals if `se` or `ci` is chosen. The default value is 3. The minimum value is 0, and the maximum is 11. If one number is specified in *numlist*, it will apply to all coefficients. If multiple numbers are specified, the first number will determine the decimals reported for the first coefficient; the second number, the decimals for the second coefficient; and so on. If there are fewer numbers in the *numlist* than coefficients, the last number in the numlist will apply to all the remaining coefficients.

`bfmt(`*textlist*`)` specifies the format type for coefficient estimates (and standard errors or confidence intervals, if `se` or `ci` is chosen). Possible format types include `e` for scientific notation (for example, `1.00e+3`), `f` or `fc` for fixed format (with commas for thousands with `fc`), and `g` or `gc` for general format (with commas for thousands with `gc`). The default for `bfmt` is `fc`. If multiple format types are specified, they are applied to the coefficients the way that multiple `bdec` parameters are applied. This option is mainly to allow scientific notation. For an explanation of Stata numeric formats, see [U] **15.5.1 Numeric formats**.

`coefastr` specifies that asterisks for significance levels are appended to regression coefficients rather than to $t$ statistics or standard errors.

### Options for statistics, standard error, etc.

`se` specifies that standard errors rather than $t$ statistics are reported.

`pvalue` specifies that $p$-values (of $t$ statistics) rather than $t$ statistics are reported.

`ci` specifies that confidence intervals of coefficients rather than $t$ statistics are reported.

`tdec(#)` specifies the number of decimal places reported for $t$ statistics (or for $p$-values if `pvalue` is specified). It also specifies the decimal places reported for $R$-squared or adjusted $R$-squared if they are not specified in `rdec`. The default value for `tdec` is usually 2, but 3 if `pvalue` is specified. The minimum value is 0, and the maximum is 11.

`noparen` specifies that no parentheses be placed around $t$ statistics or standard errors.

`bracket` specifies that square brackets be used rather than parentheses around $t$ statistics or standard errors.

`noaster` specifies that no asterisks denoting 1% and 5% significance levels be reported.

`3aster` specifies 3 asterisks for 1%, 2 asterisks for 5%, and 1 asterisk for 10% significance levels.

`10pct` specifies a plus sign for 10% significance levels in addition to the default 2 asterisks for 1%, and 1 asterisk for 5% significance levels.

`sigsymb(`*textlist*`)` specifies symbols for 1% and 5% significance levels (and 10% significance level if `10pct` is also chosen). The specified symbols replace the asterisks. Quotation marks around the new symbols are optional if the characters "," and ")" are avoided. Omitting symbols will prevent the significance level from being labeled (see also `noaster`). For example, to display only 1% significance levels, one could use `outreg using table1, sigsymb(*)`.

### Options for statistics

`nocons` specifies that the intercept (constant) coefficient estimate not be reported.

`nonobs` specifies that the number of observations in the estimation not be reported.

`noni` specifies that the number of groups in a panel data regression not be reported (for example, the number of groups specified by the `i()` variable in `xtreg`).

`nor2` specifies that no $R$-squared (or adjusted $R$-squared) be reported. This option is only meaningful when Stata calculates a true $R$-squared.

`adjr2` specifies that the adjusted $R$-squared be reported rather than the regular $R$-squared.

`rdec(#)` specifies the number of decimal places reported for the $R$-squared or adjusted $R$-squared. The default value for `rdec` is the value for `tdec`. The minimum value is 0, and the maximum is 11.

`addstat(`*text*`, #` [`, `*text*`, #, ...`]`)` specifies user-added statistics to be displayed in new lines below the $R$-squared (if shown). The user must specify both a name and a value for the statistic. Users can report significance levels of test statistics as a second statistic to be shown on the line below the first statistic.

`adec(`*numlist*`)` specifies the number of decimal places reported for user-added statistics (in `addstat`). The default value for `rdec` is the value for `tdec`. The minimum value is 0, and the maximum is 11. If one number is specified in *numlist*, it will apply to all statistics. If multiple numbers are specified in *numlist*, they are applied to the user-added statistics as in `bdec`.

`eform` specifies that the exponential form of coefficients be reported. This corresponds to the `or` option for `logit`, `clogit`, and `glogit` estimation, `irr` for `poisson` estimation, `rrr` for `mlogit`, `hr` for `cox` hazard models, and `eform` for `xtgee`, but it can be used to exponentiate the coefficients after any estimation; see *Methods and Formulas* in [R] **maximize**.

`marginal` specifies that the marginal effects rather than the coefficient estimates are reported. This is done automatically after `dprobit`.

`onecol` specifies that multi-equation models (for example, `mlogit`, `reg3`) be formatted in one column rather than the default of multiple columns, one column per equation. It also reports extra statistics included in the `e(b)` vector.

`xstats` specifies that the extra statistics included in the `e(b)` matrix be reported. Extra statistics for multi-equation models (for example, `heckman`, `heckprob`, and `biprobit`) are not reported; a user can use `addstat` or `onecol`. If there are no extra statistics in the `e(b)` matrix, `xstats` is ignored. This option is largely superseded by `addstat`.

## Other options

`comma` specifies that the ASCII file output be separated by commas rather than by tabs. This can cause problems if any of the user-defined text has commas in it (such as variable labels, `title`, `ctitle`, `addstat`, or `addnote`).

`append` specifies that new estimation output be appended to an existing output file. In general, the same `outreg` options should be used in the original regression output and each appended regression. The notes at the bottom of the table explaining the $t$ statistics or standard errors and asterisks are correct for the first estimation in the output file. If subsequently appended estimation results use different options (such as a switch to `noaster`, or changes the estimation's `robust` option), the notes will not be appropriate for all the columns. This problem can be addressed with a combination of `nonotes` and `addnote`.

`replace` specifies that it is okay to replace *filename* if it already exists.

## Examples

We begin by using Stata's automobile data.

```
. generate weight2=weight^2
. regress mpg weight weight2 foreign
  Source |       SS       df       MS                  Number of obs =      74
---------+------------------------------                F(  3,    70) =   52.25
   Model |  1689.15372     3   563.05124                Prob > F      =  0.0000
Residual |   754.30574    70  10.7757963                R-squared     =  0.6913
---------+------------------------------                Adj R-squared =  0.6781
   Total |  2443.45946    73  33.4720474                Root MSE      =  3.2827
------------------------------------------------------------------------------
     mpg |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
  weight |  -.0165729   .0039692     -4.175   0.000      -.0244892   -.0086567
 weight2 |   1.59e-06   6.25e-07      2.546   0.013       3.45e-07    2.84e-06
 foreign |    -2.2035   1.059246     -2.080   0.041        -4.3161   -.0909003
   _cons |   56.53884   6.197383      9.123   0.000       44.17855    68.89913
------------------------------------------------------------------------------

. outreg using outreg1
. !more outreg1.out
        Mileage (mpg)
Weight (lbs.)   -0.017
        (4.18)**
weight2 0.000
        (2.55)*
Car type        -2.204
        (2.08)*
Constant        56.539
        (9.12)**
Observations    74
R-squared       0.69
Absolute value of t-statistics in parentheses
* significant at 5%; ** significant at 1%
```

Next we use some of the `outreg` options.

```
. outreg  weight weight2 using outreg1, se bdec(4) bfmt(e) tdec(3) nonotes replace
. !more outreg1.out
       Mileage (mpg)
Weight (lbs.)   -1.6573e-02
       (3.9692e-03)**
weight2 1.5912e-06
       (6.2489e-07)*
Constant        5.6539e+01
       (6.1974e+00)**
Observations    74
R-squared       0.691
```

Finally, here is an example I used in my research which investigates the relationship of a dependent variable $y$ on four $X$ variables and shows how `outreg` can be used to produce the kinds of tables needed in publications.

```
. regress y x1-x2
. outreg using table4, title("Three Regression Variants")
> addnote("", "Run at $S_TIME, $S_DATE", Using data from $S_FN) replace
. regress y x1-x3
. outreg using table4, append
. regress y x1 x3-x4
. test x3 x4
. outreg using table4, append addstat("F test, x3=x4=0", r(F), Prob > F, r(p)) adec(2,3)
. type table4.out
           Three Regression Variants
                   (1)       (2)       (3)
                    y         y         y
   x1              0.686     0.723    -0.388
                  (4.47)**  (3.79)**  (0.62)
   x2             -3.781    -3.809
                 (25.66)** (22.21)**
   x3                        4.107    57.701
                            (0.33)    (1.84)
   x4                                 21.560
                                      (1.86)
   Constant         1.330     0.224   -45.005
                   (0.40)    (0.05)    (1.82)
   Observations      100       100       100
   R-squared        0.99      0.99      0.99
   F test, x3=x4=0                      1.78
   Prob > F                             0.174
   Absolute value of t-statistics in parentheses
   * significant at 5% level; ** significant at 1% level
   Run at 12:10:19, 18 Apr 2000
   Using data from xydata.dta
```

## Acknowledgments

Thanks to many Statalist users for bug reports and suggestions. Mead Over's comments were especially useful.

## References

Gallup, J. L. 1998. sg97: Formatting regression output for published tables. *Stata Technical Bulletin* 46: 28–30. Reprinted in *Stata Technical Bulletin Reprints*, vol. 8, pp. 200–202.

——. 1999. sg97.1: Revision of outreg. *Stata Technical Bulletin* 49: 23. Reprinted in *Stata Technical Bulletin Reprints*, vol. 9, pp. 170–171.

| sg153 | Censored least absolute deviations estimator: CLAD |
|---|---|

Dean Jolliffe, Economic Research Service, U.S. Department of Agriculture, jolliffe@ers.usda.gov
Bohdan Krushelnytskyy, Czech Republic, bohdan.krushelnytskyy@cerge.cuni.cz
Anastassia Semykina, Czech Republic, anastassia.semykina@cerge.cuni.cz

**Abstract:** The command `clad` for estimating Powell's (1984) censored least absolute deviations estimator and bootstrap estimates of its sampling variance is introduced and illustrated.

**Keywords:** censored least absolute value deviations estimator, bootstrap.

## Description

This insert provides the program clad for estimating Powell's (1984) censored least absolute deviations estimator (CLAD) and bootstrap estimates of its sampling variance. The CLAD estimator is a generalization of the least absolute deviations (LAD) estimator, which is implemented in Stata in the command qreg. Unlike the standard estimators of the censored regression model such as tobit or other maximum likelihood approaches, the CLAD estimator is robust to heteroscedasticity and is consistent and asymptotically normal for a wide class of error distributions. See Arabmazar and Schmidt (1981) and Vijverberg (1987) for empirical examples of the magnitude of the bias resulting from the tobit estimator in the presence of nonnormal error distributions.

This program sidesteps the issue of programming analytical standard errors and provides instead bootstrapped estimates of the sampling variance. Rogers (1993) shows that the standard errors reported by Stata for qreg are not robust to violations of homoscedasticity or independence of the residuals and proposes a bootstrap alternative. We follow Rogers for the CLAD estimator and propose two bootstrap estimates of the standard errors. The first is the standard bootstrap which assumes that the sample was selected using a simple random design. The second is a bootstrap estimate which assumes that the sample was selected in two stages and which replicates the design by bootstrapping in two stages.

An advantage of the two-stage bootstrap estimates available in clad is that if the sample was collected using a two-stage process, then the estimated standard errors will be robust to this design effect. Kish (1995) and Cochran (1997) show the importance of correcting mean values for design effects. Scott and Holt (1982) show that the magnitude of the bias for the estimated variance–covariance matrix for ordinary least squares estimates can be quite large when it is erroneously assumed that the data were collected using a simple random sample; if in fact a two-stage design had been used.

## Syntax

clad *varlist* [if *exp*] [in *range*] [, reps(#) psu(*varname*) ll[(#)] ul[(#)] dots saving(*filename*)

replace level(#) quantile(#) iterate(#) wlsiter(#) ]

## Options

reps(#) specifies the number of bootstrap replications to be performed. The default value is 100.

psu(*varname*) specifies the variable identifying the primary sampling unit. If no variable is specified, then the bootstrap replication is a single-stage, simple random draw on the sample.

ll[(#)] and ul[(#)] are as in Stata's tobit command and indicate the censoring point. ll() indicates left censoring and ul() indicates right censoring. If ll or ul is specified without a specific censoring value, then clad assumes that the lower limit is the minimum observed in the data (if ll is specified) and the upper limit is the maximum (if ul is specified). If nothing is specified for a lower or upper bound, clad assumes that the lower limit is zero. clad only functions with lower or upper censoring; one cannot specify censoring at both the lower and upper bound.

dots prints a dot to the screen for each bootstrap replication; thereby allowing the user to estimate, after a few replications, the time to completion.

saving(*filename*) creates a Stata datafile (.dta file) containing the bootstrap sample of the parameter estimates.

replace overwrites the Stata datafile specified in saving(), if it already exists.

level(#) specifies the confidence level, in percent, for confidence intervals. The default is level(95) or as set by set level.

quantile(#) specifies the quantile to be estimated and should be a number between 0 and 1, exclusive. Numbers larger than 1 are interpreted as a percent. The default value of 0.5 corresponds to the median.

iterate(#) specifies the maximum number of iterations that will be allowed to find a solution. The default value is 16,000, and the range is 1 to 16,000.

wlsiter(#) specifies the number of weighted least squares iterations that will be attempted before the linear programming iterations are started. The default value is 1. If there are convergence problems—something we have never observed—increasing this value should help.

## Examples

To illustrate the use of clad, we use data from the 1988 Ghana Living Standard Survey (GLSS) and consider a somewhat nonsensical regression. The sample considered is 1,581 households, and the dependent variable, loffinc, is the log of household, nonfarm income. Since some households are fully engaged in farming, this variable has 528 observations with zeros recorded. This variable is regressed on the log of the size of the household, lsize, and two geographic dummy variables, urban and coastal. When we issue clad we obtain the results below.

```
. clad loffinc lsize urban coastal, ll(0) reps(200)
Initial sample size = 1581
Final sample size = 1580
Pseudo R2 = .05048178
Bootstrap statistics
Variable |   Reps   Observed       Bias   Std. Err.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
   lsize |    200   1.149846   .0554115   .2544479    .6480861  1.651606  (N)
         |                                           .7073701  1.689895  (P)
         |                                           .6859084  1.624102 (BC)
---------+--------------------------------------------------------------------
   urban |    200   2.375166   .0128999   .3375226    1.709586  3.040746  (N)
         |                                           1.642076  3.120919  (P)
         |                                           1.677854  3.184893 (BC)
---------+--------------------------------------------------------------------
 coastal |    200   1.287741  -.0094159   .2830439    .7295905  1.845891  (N)
         |                                           .7311435  1.863342  (P)
         |                                           .7339153   1.90661 (BC)
---------+--------------------------------------------------------------------
   const |    200   6.443694  -.0810437   .6198413    5.221394  7.665994  (N)
         |                                           4.956254  7.557803  (P)
         |                                           5.371459  7.730506 (BC)
------------------------------------------------------------------------------
                           N = normal, P = percentile, BC = bias-corrected
```

The first line of output tells us that the original sample size is 1,581 and in the second line we learn that the algorithm for estimation dropped one case from the sample. An important caveat to the pseudo $R$-squared reported on the third line, is that this is the reported statistic from the last iteration of the `qreg` command on the final sample size. It is not the pseudo $R$-squared for the original sample, but we have opted to report this statistic to provide some indication of how the model is performing.

In the example above, no sample design information is passed to `clad` and the program calls Stata's `bsample` utility to resample the data 200 times. In order to maintain the same sample size in each bootstrap resample, `clad` ignores observations where the dependent variable is missing. The results from `bsample` are then passed to the `bstat` command to generate the standard Stata bootstrap output. For more information about the normal, percentile, and bias-corrected percentile confidence intervals, see `bstrap` in the Stata manuals. For an introduction to the bootstrap principle, see Efron and Tibshirani (1993). In order to reproduce results from `clad`, it is necessary first to set the random number seed; see `generate` in the Stata reference manuals for more information.

The reported standard errors above will be correct if the sample comes from a simple random draw. This is not the case with the GLSS data, which was collected using a two-stage design. `clad` can generate bootstrap estimates of the standard errors which are robust to the two-stage design by passing the information about the primary sampling unit (PSU) to `clad`. For example, we correct the standard errors above for this aspect of the sample in the example below.

```
. clad loffinc lsize urban coastal, ll(0) reps(200) psu(clust)
Initial sample size = 1581
Final sample size = 1580
Pseudo R2 = .05048178
Bootstrap statistics
Variable |   Reps   Observed       Bias   Std. Err.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
   lsize |    200   1.149846   .0916958    .395014    .3708959  1.928797  (N)
         |                                           .6573149  2.076703  (P)
         |                                           .6507832  2.053507 (BC)
---------+--------------------------------------------------------------------
   urban |    200   2.375166   .0562143   .6152112    1.161996  3.588336  (N)
         |                                           1.285434  3.658858  (P)
         |                                            1.12299  3.495041 (BC)
---------+--------------------------------------------------------------------
 coastal |    200   1.287741   .0386539   .5439033    .2151873  2.360294  (N)
         |                                           .2898641  2.466994  (P)
         |                                           .0728349  2.216781 (BC)
---------+--------------------------------------------------------------------
   const |    200   6.443694  -.1804084    1.04149    4.389922  8.497466  (N)
         |                                           3.942665  8.130428  (P)
         |                                           4.440762  8.347237 (BC)
------------------------------------------------------------------------------
                           N = normal, P = percentile, BC = bias-corrected
```

It is worth noting that introducing information about the sample design only affects the estimates of the standard errors.

The dramatic increase in the size of the standard errors is not that surprising as the design effect for the dependent variable is approximately 3.8, and there is little in the observation matrix which will explain the intracluster correlation.

## Methods and Formulas

The Powell (1984) CLAD estimator is found by minimizing

$$\sum |y_i - \max(0, x_i'\beta)| \tag{1}$$

The consistency of this estimator rests on the fact that medians are preserved by monotone transformations of the data, and (1) is a monotone transformation of the standard least absolute deviations (LAD) regression. The properties of the LAD estimator are presented in Koenker and Basset (1978). The LAD estimator is implemented in Stata with the `qreg` command.

The estimation technique used in `clad` for the CLAD estimator is Buchinsky's (1991) iterative linear programming algorithm (ILPA). (For a critique of and alternative to this algorithm, see Fitzenberger 1997.) The first step of the ILPA is to estimate a quantile regression for the full sample, then delete the observations for which the predicted value of the dependent variable is less than zero. Another quantile regression is estimated on the new sample, and again negative predicted values are dropped. More generally, observations are dropped if the predicted value is less than the censoring value when the left tail of the distribution is censored, or they are dropped if the predicted value is greater than the censoring value when the right tail of the distribution is censored. Buchinsky (1991) shows that if the process converges, then a local minimum is obtained. Convergence occurs when there are no negative predicted values in two consecutive iterations.

The two bootstraps are implemented as follows. For the simple random sample (SRS) we simply use Stata's `bsample` utility to bootstrap the CLAD point estimates. The SRS, two-stage bootstrap follows this process. In the first stage it counts the number of unique PSUs, say $k$, and then using Stata's `uniform` function, randomly selects with replacement $k$ (not necessarily unique) PSUs. At this point, it counts the number of times each PSU has been selected, and this is stored for later use. To implement the second stage, the program first counts the number of ultimate sampling units (USUs), say $m$, in each selected PSU and then randomly selects $m$ USUs from each selected PSU. If a PSU is selected more than once, say $\alpha$ times, then in the second stage the program randomly selects $\alpha m$ USUs from the selected PSU. As a final note, we warn that `clad` can be quite time consuming since the entire algorithm described above is repeated for each bootstrap resampling of the data.

## References

Arabmazar, A. and P. Schmidt. 1981. Further evidence on the robustness of the tobit estimator to heteroskedasticity. *Journal of Econometrics* 17: 253–258.

Buchinsky, M. 1991. Methodological issues in quantile regression, Chapter 1 of *The Theory and Practice of Quantile Regression* Ph.D. dissertation, Harvard University.

——. 1994. Changes in the U.S. wage structure 1963–1987: application of quantile regression. *Econometrica* 62(2): 405–459.

Cochran, W., 1997. *Sampling Techniques*. 3d ed. New York: John Wiley & Sons.

Efron, B. and R. Tibshirani. 1993. *An Introduction to the Bootstrap, Monographs on Statistics and Applied Probability*. New York: Chapman & Hall.

Fitzenberger, B. 1997. Computational aspects of censored quantile regression. In *Proceedings of The 3rd International Conference on Statistical Data Analysis based on the L1 B Norm and Related Methods*, ed. Y. Dodge, 171–186. Hayword, California: Institute of Mathematical Statistics Lecture Notes B Monograph Series, Volume 31.

Kish, L. 1995. *Survey Sampling*. New York: John Wiley & Sons.

Koenker, R. and G. Bassett. 1978. Regression quantiles. *Econometrica* 46(1): 33–50.

Powell, J. L. 1984. Least absolute deviations estimation for the censored regression model. *Journal of Econometrics* 25: 303–325.

Rogers, W. 1993. sg11.2: Calculation of quantile regression standard errors. *Stata Technical Bulletin* 13: 18–19. Reprinted in *Stata Technical Bulletin Reprints*, vol. 3, pp. 77–78.

Scott, A. J. and D. Holt. 1982. The effect of two-stage sampling on ordinary least squares methods. *Journal of the American Statistical Association* 77(380): 848–854.

Vijverberg, W. 1987. Non-normality as distributional misspecification in single-equation limited dependent variable models. *Oxford Bulletin of Economics and Statistics* 49(4): 417–430.

| sg154 | Confidence intervals for the ratio of two binomial proportions by Koopman's method |
|---|---|

Duolao Wang, London School of Hygiene and Tropical Medicine, London, UK, duolao.wang@lshtm.ac.uk

**Abstract:** This article introduces the `koopman` and `koopmani` commands, which compute confidence intervals for the ratio of two binomial proportions based on two independent binomially distributed random variables using Koopman's method.

**Keywords:** Koopman's method, odds ratio, confidence intervals.

## Koopman's method

Let $X$ and $Y$ be two independent binomial variates based on sample sizes $m$ and $n$ and parameters $p_1$ and $p_2$, respectively. Let $\theta = p_1/p_2$. Koopman (1984) proposed a method for constructing confidence intervals for $\theta$ based on a chi-squared test. Koopman's method has been widely used in medical research for evaluating drug efficacy and treatment effects.

Assume we test for $H_0 : \theta = \theta_0$ against $H_A : \theta \neq \theta_0$. For this problem there is no uniformly most powerful test as extreme values may occur in the sample, but a chi-squared test seems a reasonable choice. The test statistic $U_{\theta_0}$ is then given by

$$U_{\theta_0}(x, y) = \frac{(x - m\widehat{p}_1)^2}{m\widehat{p}_1(1 - \widehat{p}_1)} + \frac{(y - n\widehat{p}_2)^2}{n\widehat{p}_2(1 - \widehat{p}_2)}$$

where $\widehat{p}_1$ and $\widehat{p}_2$ are the maximum likelihood estimates under the restriction $\theta = \theta_0$. It can be proved that

$$\widehat{p}_1 = \frac{\theta_0(m + y) + x + n - \left[\{\theta_0(m + y) + x + n\}^2 - 4\theta_0(m + n)(x + y)\right]^{1/2}}{2(m + n)}$$

and $\widehat{p}_2 = \widehat{p}_1/\theta_0$.

For $\theta = 1$, the statistic $U_{\theta_0}(x, y)$ is the traditional Pearson chi-square. Rearranging $U_{\theta_0}(x, y)$ results in

$$U_{\theta_0}(x, y) = \frac{(x - m\widehat{p}_1)^2}{m\widehat{p}_1(1 - \widehat{p}_1)} \left\{1 + \frac{m(\theta_0 - \widehat{p}_1)}{n(1 - \widehat{p}_1)}\right\}$$

This shows that under $H_0$, $U_{\theta_0}(x, y)$ has asymptotically for $m \to \infty$ and $n \to \infty$, a chi-squared distribution with 1 degree of freedom independent of $\theta_0$ (Bishop et al. 1977). Hence, an approximate $1 - \alpha$ two-sided confidence region for $\theta$ is given by

$$\{U_{\theta_0}(x, y) < \chi^2_{1,1-\alpha}\}$$

where $\chi^2_{1,1-\alpha}$ is the $1 - \alpha$ fractile of the chi-squared distribution with 1 degree of freedom. Since $U$ is a convex function of $\theta$, this is an asymmetric interval $(\theta_l, \theta_u)$, where

$$U_{\theta_l}(x, y) = U_{\theta_u}(x, y) = \chi^2_{1,1-\alpha}$$

and

$$\theta_l < \theta_u$$

As $U_{\theta_l}(x, y)$ reduces to the usual chi-squared when $\theta = 1$, this interval will always agree with the chi-squared test.

Because there is no explicit expression for the inverse function of $U$, the values of $\theta_l$ and $\theta_u$ have to be solved by numerical procedures. The main concern of the command `koopman` is to obtain $\theta_l$ and $\theta_u$ by using repeated bisection as suggested by Koopman (1984).

## Syntax

`koopman` *var_event var_group* [*weight*] [`if` *exp*] [`in` *range*] [`, level(#)`]

`koopmani` $\#_x$ $\#_m$ $\#_y$ $\#_n$ [`, level(#)`]

`koopman` allows `fweights`.

## Description

`koopman` computes confidence intervals for the ratio of two binomial proportions based on two independent binomially distributed random variables using Koopman's method. Point estimates and confidence intervals for the odds ratio are calculated. *event_var* contains a one if the observation represents an event and 0 otherwise. *group_var* indicates the group to which each observation belongs. The variable must have only two values. Observations with missing values are not used.

`koopmani` is the immediate form of `koopman`.

## Options

`level(#)` specifies the confidence level, in percent, for confidence intervals. The default is `level(95)` or as set by `set level`.

## Examples

The likelihood ratio that a diagnostic procedure will detect a certain disease is defined as the ratio of the fraction of true positives to the fraction of false positives. To determine this likelihood ratio, $\theta$, for which mainly high values are of interest, $m = 40$ diseased persons were tested from whom $x = 36$ had a positive test, and $n = 80$ nondiseased persons were tested from whom $y = 16$ had a positive test. We can now use `koopman` to compute the point estimate of $\theta$ and its confidence intervals.

```
. koopmani 36 40 16 80
                 |            Event        |            Proportion
                 |       Yes          No   |     Total         Yes
-----------------+-------------------------+----------------------
          Group1 |        36           4   |        40      0.9000
          Group2 |        16          64   |        80      0.2000
-----------------+-------------------------+----------------------
           Total |        52          68   |       120      0.4333
                 |                         |
                 |      Point estimate     |  [95% Conf. Interval]
-----------------|-------------------------+----------------------
      Odds Ratio |               4.5       |  2.939633    7.152252
------------------------------------------------------------------

. koopmani 36 40 16 80, level(99)
                 |            Event        |            Proportion
                 |       Yes          No   |     Total         Yes
-----------------+-------------------------+----------------------
          Group1 |        36           4   |        40      0.9000
          Group2 |        16          64   |        80      0.2000
-----------------+-------------------------+----------------------
           Total |        52          68   |       120      0.4333
                 |                         |
                 |      Point estimate     |  [99% Conf. Interval]
-----------------|-------------------------+----------------------
      Odds Ratio |               4.5       |  2.598043    8.283284
------------------------------------------------------------------
```

`koopman` works like `koopmani` except that it obtains the entries in the tables by summing data. We specify three variables. The first represents whether the event occurred, the second represents which group the observation belongs to, and the third variable a weight that gives the total number of subjects in this observation. An observation may reflect a single subject or a group of subjects. If an observation represents a single subject, we have the individual data in which the weight for each subject is 1.

```
. clear
. use koopman
. list
       event     group        pop
  1.        1         1         36
  2.        0         1          4
  3.        1         2         16
  4.        0         2         64
. koopman event group [freq=pop], level(90)
                 |            Event        |            Proportion
                 |       Yes          No   |     Total         Yes
-----------------+-------------------------+----------------------
          Group1 |        36           4   |        40      0.9000
          Group2 |        16          64   |        80      0.2000
-----------------+-------------------------+----------------------
           Total |        52          68   |       120      0.4333
                 |                         |
                 |      Point estimate     |  [90% Conf. Interval]
-----------------|-------------------------+----------------------
      Odds Ratio |               4.5       |  3.136696    6.632661
------------------------------------------------------------------
```

## Saved results

koopman and koopmani save in r():

Scalars

| | |
|---|---|
| r(x) | number of events in group 1 |
| r(m) | total number of subjects in group 1 |
| r(y) | number of events in group 2 |
| r(n) | total number of subjects in group 2 |
| r(theta) | odds ratio between group 1 and 2 |
| r(theta_l) | lower bound of CI for theta |
| r(theta_u) | upper bound of CI for theta |

## References

Bishop, Y. M. M., S. E. Fienberg, and P. W. Holland. 1977. *Discrete Multivariate Analysis.* Cambridge, MA: MIT Press.

Koopman, P. A. R. 1984. Confidence intervals for the ratio of two binomial proportions. *Biometrics* 40: 513–517.

# sg155 | Tests for the multinomial logit model

Jeremy Freese, University of Wisconsin-Madison, jfreese@ssc.wisc.edu
J. Scott Long, Indiana University, jslong@indiana.edu

**Abstract:** The command mlogtest designed to simplify the use of several tests associated with the multinomial logit model is introduced and illustrated.

**Keywords:** multinomial logit model, Hausman, likelihood-ratio test, Wald test.

## Introduction

There are several tests that are commonly used in association with the multinomial logit model (MNLM hereafter). First, we can test that all of the coefficients associated with an independent variable are simultaneously equal to zero (that is, test that a variable has no effect). Second, we can test whether the independent variables differentiate between two outcomes; this test is commonly used to determine if two outcomes can be combined. Third, we can assess the assumption of the independence of irrelevant alternatives (IIA) using either a Hausman test or the LR test proposed by McFadden et al. (1976) and improved by Small and Hsiao (1985). While each of these tests can be computed using either the test, lrtest, or hausman commands in Stata or the smhsiao command of Nick Winter (available at the SSC-IDEAS archive), in practice computing these tests can be awkward and/or tedious. The mlogtest command is designed to simplify the use of these tests. mlogtest is a post-estimation command that requires that mlogit is the last model estimated.

Given the difficulties of interpretation associated with the MNLM, it is tempting to search for a more parsimonious model by excluding variables or combining outcome categories based on a series of statistical tests. While mlogtest facilitates computing tests that can be used in a specification search, great care is required. First, these tests all involve multiple coefficients. While the overall test might indicate that *as a group* the parameters are not significantly different from zero, an *individual* parameter can still be substantively and statistically significant. Accordingly, one needs to carefully examine the individual coefficients involved in each test before deciding to revise a model. Second, as with all searches that use repeated, sequential tests, there is a danger of overfitting the data. When models are constructed based on prior testing using the same data, significance levels should only be used as rough guidelines.

## Syntax

mlogtest [, detail iia hausman lr wald combine lrcomb smhsiao set(*varlist* [\ *varlist...*]) all base ]

## Options

detail reports the full hausman output for the IIA test. The default is to provide only a summary of the results.

iia specifies that both tests of the IIA assumption should be performed.

hausman requests Hausman tests of the IIA assumption.

lr requests that LR tests for each independent variable should be performed.

wald requests that Wald tests for each independent variable should be performed.

combine requests Wald tests of whether dependent categories can be combined.

`lrcomb` requests LR tests of whether dependent categories can be combined. This option uses constrained estimation and overwrites constraint 999 if it is already defined.

`smhsiao` requests Small–Hsiao tests of the IIA assumption.

`set(`*varlist* [\ *varlist*...]`)` specifies that a set of variables is to be considered together for the LR test or Wald test. The backslash is used to specify multiple sets of variables. This option is particularly useful when a categorical independent variable is entered as a set of dummy variables.

`all` requests that all available tests should be performed.

`base` also conducts an IIA test omitting the base category of the original `mlogit` estimation. This is done by reestimating the model using the largest remaining category as the base category, although the original estimates are restored to memory afterward.

## Utility procedures

`mlogtest` uses several utility ado-files that are also used in other programs by the authors. In this section we briefly describe these ado-files.

`_perhs.ado` returns the number of right-hand-side variables and their names for regression models.

`_pecats.ado` returns the names and values of the categories for models with ordinal, nominal, or binary outcomes. For `mlogit` it indicates the value of the reference category.

## Example

The data for this example are from the 1993 and 1994 General Social Survey. The nominal variable (`kidvalue`) is the respondent's choice of which of the following is most important for a child to learn to prepare him or her for life: "to obey" (`kidvalue` = 1), "to think for himself or herself" (`kidvalue` = 2), "to work hard" (`kidvalue` = 3), or "to help others when they need help" (`kidvalue` = 4). The fifth option, "to be popular", was excluded because it was very rarely chosen. The independent variables are respondent's sex (`female`), race (`black` and `othrrace`, with the reference category being `white`), education (`degree`), and whether the respondent has any children of her or his own (`anykids`). We begin by estimating the MNLM.

```
. mlogit kidvalue female black othrrace degree anykids, nolog
Multinomial regression                          Number of obs   =       2978
                                                LR chi2(15)     =     300.14
                                                Prob > chi2     =     0.0000
Log likelihood = -3396.3518                     Pseudo R2       =     0.0423
------------------------------------------------------------------------------
kidvalue |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
obey     |
  female |  -.2605371   .1048637     -2.485   0.013      -.4660662   -.0550079
   black |   .3297048   .1452035      2.271   0.023       .0451112    .6142984
othrrace |   .5711209   .2872073      1.989   0.047       .0082049    1.134037
  degree |  -.7040498   .0577797    -12.185   0.000       -.817296   -.5908037
 anykids |  -.0401693   .1202552     -0.334   0.738      -.2758652    .1955265
   _cons |  -.0847716   .1376452     -0.616   0.538      -.3545513    .1850081
---------+--------------------------------------------------------------------
workhard |
  female |  -.4657661   .1104007     -4.219   0.000      -.6821476   -.2493846
   black |   .1975939   .1714529      1.152   0.249      -.1384475    .5336354
othrrace |   1.621659   .2233146      7.262   0.000       1.183971    2.059348
  degree |  -.1824923   .0479872     -3.803   0.000      -.2765455   -.0884391
 anykids |   .0052844   .1243124      0.043   0.966      -.2383635    .2489323
   _cons |  -.8719322   .1472885     -5.920   0.000      -1.160612   -.5832521
---------+--------------------------------------------------------------------
helpoth  |
  female |  -.3530656   .1165728     -3.029   0.002      -.5815441   -.1245871
   black |  -.1156104   .1892914     -0.611   0.541      -.4866148     .255394
othrrace |   .8759096   .2791998      3.137   0.002        .328688    1.423131
  degree |  -.3875589   .0549027     -7.059   0.000      -.4951661   -.2799517
 anykids |  -.1913028   .1286881     -1.487   0.137      -.4435269    .0609214
   _cons |  -.5615834   .1493388     -3.760   0.000      -.8542821   -.2688846
------------------------------------------------------------------------------
(Outcome kidvalue==thnkself is the comparison group)
```

In the following examples, we use a series of `mlogtest` commands to estimate several tests. Alternatively, we could have

requested any combination of tests by combining options or requested all possible tests with the single command: `mlogtest, all`.

### Tests of independent variables

We first conduct a LR test for each independent variable.

```
. mlogtest, lr
**** Likelihood-ratio tests for independent variables
 Ho: All coefficients associated with given variable(s) are 0.
kidvalue |      chi2    df   P>chi2
---------+--------------------------
  female |     23.558    3    0.000
   black |      7.231    3    0.065
othrrace |     51.944    3    0.000
  degree |    211.133    3    0.000
 anykids |      2.323    3    0.508
-----------------------------------
```

For example, we can reject the hypothesis that gender does not affect the values considered important for children at the .01 level, or the effect of gender is significant ($p < .01$). Next, we conduct a Wald test for each independent variable. We also use the `set` option to test the hypothesis that the coefficients for the two dummy variables indicating race are simultaneously equal to zero.

```
. mlogtest, wald set(black othrrace)
**** Wald tests for independent variables
 Ho: All coefficients associated with given variable(s) are 0.
kidvalue |       chi2   df   P>chi2
---------+--------------------------
  female |     23.451    3    0.000
   black |      7.317    3    0.062
othrrace |     54.177    3    0.000
  degree |    174.002    3    0.000
 anykids |      2.359    3    0.501
---------+----------------------------
set_1:   |     60.988    6    0.000
   black |
othrrace |
-----------------------------------
```

### Tests of IIA

Either the Hausman or Small–Hsiao tests can be used to test the IIA assumption. We begin with the Hausman test. The `base` option specifies that all tests should be computed using the most frequently observed remaining category as the base value; see *Methods and formulas* for details. We do not use the `detail` option, which provides all of the output from the successive calls to Stata's `hausman` command.

```
. mlogtest, hausman base
**** Hausman tests of IIA assumption
 Ho: Odds(Outcome-J vs Outcome-K) are independent of other alternatives.
 Omitted |      chi2   df   P>chi2   evidence
---------+-----------------------------------
    obey |     7.764   12    0.803   for Ho
workhard |    -4.090   12     ---    for Ho
 helpoth |     9.154   12    0.690   for Ho
thnkself |   884.043   12    0.000   against Ho
-------------------------------------------------
 Note: If chi2<0, the estimated model does not
 meet asymptotic assumptions of the test.
```

Note the considerably different results depending on the category considered. In our experience, negative test statistics are very common; Hausman and McFadden (1984, 1226) note this possibility and conclude that a negative result is evidence that IIA has *not* been violated. When we run Small–Hsiao tests, we see that these results vary considerably from those of the Hausman tests.

```
. mlogtest, smhsiao base
```

```
**** Small-Hsiao tests of IIA assumption
 Ho: Odds(Outcome-J vs Outcome-K) are independent of other alternatives.
 Omitted |  lnL(full)  lnL(omit)    chi2   df   P>chi2   evidence
---------+--------------------------------------------------------
    obey |  -1041.535  -1039.193   4.683    6    0.585   for Ho
workhard |  -1107.167  -1103.476   7.381    6    0.287   for Ho
 helpoth |  -1178.179  -1175.128   6.101    6    0.412   for Ho
thnkself |   -744.697   -740.162   9.069    6    0.170   for Ho
-------------------------------------------------------------------
```

Since the Small–Hsiao test is based on the creation of random half-samples from one's data, the test may differ substantially with successive calls of the command. For example, when we run the tests again, we obtain

```
. mlogtest, smhsiao base
**** Small-Hsiao tests of IIA assumption
 Ho: Odds(Outcome-J vs Outcome-K) are independent of other alternatives.
 Omitted |  lnL(full)  lnL(omit)    chi2   df   P>chi2   evidence
---------+--------------------------------------------------------
    obey |  -1098.851  -1089.556  18.589    6    0.005   against Ho
workhard |  -1164.440  -1153.210  22.459    6    0.001   against Ho
 helpoth |  -1169.482  -1165.634   7.695    6    0.261   for Ho
thnkself |   -786.601   -774.531  24.141    6    0.000   against Ho
-------------------------------------------------------------------
```

The `set seed` command can be used before `mlogtest` in a do-file to have it produce the same results with each successive run. For example, `set seed 339487731`.

### Tests for combining dependent categories

Finally, we test whether the independent variables differentiate pairs of outcome categories using a Wald test. Note that all pairs of outcomes have been evaluated.

```
. mlogtest, combine
**** Wald tests for combining outcome categories
 Ho: All coefficients except intercepts associated with given pair
     of outcomes are 0 (i.e., categories can be collapsed).
Categories tested |      chi2   df   P>chi2
------------------+------------------------
    obey-workhard |    81.629    5    0.000
    obey- helpoth |    31.332    5    0.000
    obey-thnkself |   167.265    5    0.000
workhard- helpoth |    19.637    5    0.001
workhard-thnkself |    79.317    5    0.000
 helpoth-thnkself |    65.716    5    0.000
-------------------------------------------
```

Alternatively, LR tests can be computed with the `lrcomb` option as in the next example.

```
. mlogtest, lrcom
**** LR tests for combining outcome categories
 Ho: All coefficients except intercepts associated with given pair
     of outcomes are 0 (i.e., categories can be collapsed).
Categories tested |      chi2   df   P>chi2
------------------+------------------------
    obey-workhard |    89.431    5    0.000
    obey- helpoth |    32.089    5    0.000
    obey-thnkself |   212.672    5    0.000
workhard- helpoth |    20.523    5    0.001
workhard-thnkself |    80.259    5    0.000
 helpoth-thnkself |    70.485    5    0.000
-------------------------------------------
```

As with the Wald and LR tests for each independent variable, the two tests for combining categories generally provide very similar results, although many researchers prefer the LR test.

Overall, these examples illustrate that `mlogtest` makes it very simple to compute many tests. At the risk of repetition, we note that it is not our intention to encourage researchers to combine categories or delete variables without careful consideration of the substantive issues related to the research.

## Saved results

`mlogtest` saves in `r()`:

`r(combine)` contains results of tests to combine categories. Rows represent all contrasts among categories; columns indicate the categories contrasted, the chi-squared value, the degrees of freedom, and the $p$-value.

`r(iia)` contains results of the Hausman test of IIA assumption. Each row is one test. Columns indicate the omitted category of a given test, the chi-squared value, the degrees of freedom, and the $p$-value.

`r(wald)` contains results of the Wald test that all coefficients of an independent variable equal zero.

`r(lrtest)` contains results of the LR test that all coefficients associated with an independent variable equal zero.

## Methods and formulas

This section provides brief descriptions of each of the tests. For further details, check the Stata manual for `mlogit`, `test`, and `hausman`. Full details along with citations to original sources are found in Long (1997). To make our discussion of the tests clear, we begin with a brief summary of the multinomial logit model (MNLM).

### The multinomial logit model

For simplicity, we consider a model with three outcomes and three independent variables. The MNLM can be thought of as simultaneously estimating binary logits among all pairs of the outcome categories. For example, with categories $A$, $B$, and $C$ and independent variables $x_1$, $x_2$, and $x_3$, the MNLM is in effect simultaneously estimating three binary models:

$$\ln\left[\frac{P(A|x)}{P(C|x)}\right] = \beta_{0,A|C} + \beta_{1,A|C}x_1 + \beta_{2,A|C}x_2 + \beta_{3,A|C}x_3$$

$$\ln\left[\frac{P(B|x)}{P(C|x)}\right] = \beta_{0,B|C} + \beta_{1,B|C}x_1 + \beta_{2,B|C}x_2 + \beta_{3,B|C}x_3$$

$$\ln\left[\frac{P(A|x)}{P(B|x)}\right] = \beta_{0,A|B} + \beta_{1,A|B}x_1 + \beta_{2,A|B}x_2 + \beta_{3,A|B}x_3$$

Note that three more equations could be listed, comparing $C$ to $A$, $C$ to $B$, and $B$ to $A$. Given that the sum of the probabilities for the outcomes must equal one, there is an implicit constraint on the three logits. Specifically,

$$\ln\left[\frac{P(A|x)}{P(C|x)}\right] - \ln\left[\frac{P(B|x)}{P(C|x)}\right] = \ln\left[\frac{P(A|x)}{P(B|x)}\right]$$

in terms of the parameters

$$\beta_{k,A|C} - \beta_{k,B|C} = \beta_{k,A|B}$$

`mlogit` estimates and prints only the nonredundant coefficients, which are determined by the `basecategory()` option or, by default, the category with the largest number of cases. The commands `mcross` (Rogers 1995) and `listcoef` (Long and Freese 2000) list coefficients for all comparisons of outcome categories.

### Testing the effect of an independent variable

With $J$ dependent categories, there are $J-1$ nonredundant coefficients associated with each independent variable $x_k$. The hypothesis that $x_k$ does not affect the dependent variable can be written as

$$H_0 : \beta_{k,1|\text{Base}} = \cdots = \beta_{k,J|\text{Base}} = 0$$

where Base is the base category used in the comparison. Since $\beta_{k,\text{Base}|\text{Base}}$ is necessarily zero, the hypothesis imposes constraints on $J-1$ parameters. This hypothesis can be tested with either a Wald or a LR test.

### A LR test

First, estimate the full model $M_F$ that contains all of the variables, with the resulting LR statistic $\text{LR}^2_\text{F}$. Second, estimate the restricted model $M_R$ formed by excluding variable $x_k$, with the resulting LR statistic $\text{LR}^2_\text{R}$. This model has $J-1$ fewer parameters. Finally, compute the difference $\text{LR}^2_\text{RvsF} = \text{LR}^2_\text{F} - \text{LR}^2_\text{R}$ which is distributed as chi-squared with $J-1$ degrees of freedom if the hypothesis that $x_k$ does not affect the outcome is true. `mlogtest, lr` computes this test for each of the $K$ independent variables by making repeated calls to Stata's `lrtest`. Note that this requires estimating $K$ additional multinomial logit models.

**A Wald test**

While the LR test is generally considered to be superior, if the model is complex or the sample is very large, the computational costs of the LR test can be prohibitive. Alternatively, $K$ Wald tests can be computed without estimating additional models. This test is defined as follows. Let $\widehat{\beta}_k$ be the $J-1$ coefficients associated with $x_k$. Let $\widehat{\text{Var}}(\widehat{\beta}_k)$ be the estimated covariance matrix. The Wald statistic for the hypothesis that all of the coefficients associated with $x_k$ are simultaneously zero is computed as

$$W_k = \widehat{\beta}_k' \widehat{\text{Var}}(\widehat{\beta}_k) \widehat{\beta}_k$$

If the null hypothesis is true, then $W_k$ is distributed as chi-squared with $J-1$ degrees of freedom.

**Testing multiple independent variables**

This logic of the Wald or LR tests can be extended to simultaneously test that the effects of two or more independent variables are zero. For example, the hypothesis to test that $x_k$ and $x_\ell$ have no effects is

$$H_0 : \beta_{k,1|\text{Base}} = \cdots = \beta_{k,J|\text{Base}} = \beta_{\ell,1|\text{Base}} = \cdots = \beta_{\ell,J|\text{Base}} = 0$$

The `set` option in `mlogtest` specifies which variables are to be simultaneously tested. This is particularly useful when a series of dummy variables are used to code a nominal or ordinal independent variable.

**Testing that two outcomes can be combined**

If none of the $x_k$'s significantly affect the odds of outcome $m$ versus outcome $n$, we say that $m$ and $n$ are *indistinguishable* with respect to the variables in the model. If $\beta_{1,m|n}, \ldots, \beta_{K,m|n}$ are the coefficients for $x_1$ through $x_K$ from the `logit` of $m$ versus $n$, then the hypothesis that outcomes $m$ and $n$ are indistinguishable corresponds to

$$H_0 : \beta_{1,m|n} = \cdots \beta_{K,m|n} = 0$$

Note that if the base category used by Stata is not $n$, these coefficients are not directly available. However, this hypothesis can be rewritten equivalently using the coefficients with respect to the base category

$$H_0 : \left(\beta_{1,m|\text{Base}} - \beta_{1,n|\text{Base}}\right) = \cdots = \left(\beta_{K,m|\text{Base}} - \beta_{K,m|\text{Base}}\right) = 0$$

A Wald test for this hypothesis can be computed with Stata's `test` command. `mlogtest, combine` executes and summarizes the results of $J \times (J-1)$ calls to `test` for all pairs of outcome categories.

An LR test of this hypothesis can be computed by first estimating the full model that contains all of the variables, with the resulting LR statistic $\text{LR}^2_\text{F}$. Then estimate a restricted model $M_R$ in which category $m$ is used as the base category and all the coefficients (except the constant) in the equation for category $n$ are constrained to 0, with the resulting chi-squared statistic $\text{LR}^2_\text{R}$. The test statistic is the difference $\text{LR}^2_\text{RvsF} = \text{LR}^2_\text{F} - \text{LR}^2_\text{R}$, which is distributed as chi-squared with $K$ degrees of freedom. `mlogtest, lrcomb` summarizes the results of the $J \times (J-1)$ LR tests for all pairs of outcome categories.

**Independence of irrelevant alternatives**

The MNLM assumes that the odds for any pair of outcomes are determined without reference to the other outcomes that might be available. This is known as the *independence of irrelevant alternatives* property or simply IIA. Hausman and McFadden (1984) proposed a Hausman-type test of this hypothesis. Basically, this involves the following steps.

1. Estimate the full model with all $J$ outcomes included; these estimates are contained in $\widehat{\beta}_\text{F}$.
2. Estimate a restricted model by eliminating one or more outcome categories; these estimates are contained in $\widehat{\beta}_\text{R}$.
3. Let $\widehat{\beta}^*_\text{F}$ be a subset of $\widehat{\beta}_\text{F}$ after eliminating coefficients not estimated in the restricted model. The Hausman test of IIA is defined as

$$H_\text{IIA} = \left(\widehat{\beta}_\text{R} - \widehat{\beta}^*_\text{F}\right)' \left[\widehat{\text{Var}}(\widehat{\beta}_\text{R}) - \widehat{\text{Var}}(\widehat{\beta}^*_\text{F})\right]^{-1} \left(\widehat{\beta}_\text{R} - \widehat{\beta}^*_\text{F}\right)$$

   $H_\text{IIA}$ is asymptotically distributed as chi square with degrees of freedom equal to the number of rows in $\widehat{\beta}_\text{R}$ if IIA is true. Significant values of $H_\text{IIA}$ indicate that the IIA assumption has been violated.

Hausman and McFadden (1984, 1226) note that $H_\text{IIA}$ can be negative when $\widehat{\text{Var}}(\widehat{\beta}_\text{R}) - \widehat{\text{Var}}(\widehat{\beta}^*_\text{F})$ is not positive semidefinite and suggest that a negative $H_\text{IIA}$ is evidence that IIA *holds*.

To compute Small and Hsiao's test, the sample is divided into two random subsamples of approximately equal size. The unrestricted MNLM is estimated on both subsamples. The weighted average of the coefficients from the two samples is defined as follows:

$$\widehat{\beta}_u^{S_1S_2} = \left(\frac{1}{\sqrt{2}}\right)\widehat{\beta}_u^{S_1} + \left[1 - \frac{1}{\sqrt{2}}\right]\widehat{\beta}_u^{S_2}$$

where $\widehat{\beta}_u^{S_1}$ is a vector of estimates from the unrestricted model on the first subsample and $\widehat{\beta}_u^{S_2}$ is its counterpart for the second subsample. Next, a restricted sample is created from the second subsample by eliminating all cases with a chosen value of the dependent variable. The MNLM is estimated using the restricted sample yielding the estimates $\widehat{\beta}_r^{S_2}$ and the likelihood $L(\widehat{\beta}_r^{S_2})$. The Small–Hsiao statistic is the difference:

$$SH = -2\left[L(\widehat{\beta}_u^{S_1S_2}) - L(\widehat{\beta}_r^{S_2})\right]$$

$SH$ is asymptotically distributed as a chi-squared with the degrees of freedom equal to $K + 1$, where $K$ is the number of independent variables.

For both the Hausman test and the Small–Hsiao test, multiple tests of IIA are possible. Assuming that the MNLM is estimated with base category Base, $J - 1$ tests can be computed by excluding each of the remaining categories to form the restricted model. By changing the base category, a test can also be computed that excludes Base. Note that results differ depending on which base category was used to estimate the model.

## Acknowledgments

For information on related programs and future updates to this program, please check www.indiana.edu/~jsl650/post.htm

## References

Hausman, J. A. and D. McFadden. 1984. Specification tests for the multinomial logit model. *Econometrica* 52: 1219–1240.

Long, J. S. 1997. *Regression Models for Categorical and Limited Dependent Variables.* Thousand Oaks, CA: Sage.

Long, J. S. and J. Freese. 2000. sg152: Listing and interpreting transformed coefficients from certain regression models. *Stata Technical Bulletin* 57: 27–34.

McFadden, D., W. Tye, and K. Train. 1976. An application of diagnostic tests for the independence from irrelevant alternatives property of the multinomial logit model. *Transportation Research Board Record* 637: 39–45.

Rogers, W. H. 1995. sqv10: Expanded multinomial comparisons. *Stata Technical Bulletin* 23: 26–28. Reprinted in *Stata Technical Bulletin Reprints*, vol. 4, pp. 181–183.

Small, K. A. and C. Hsiao. 1985. Multinomial logit specification tests. *International Economic Review* 26: 619–627.

# sg156 | Mean score method for missing covariate data in logistic regression models

Marie Reilly, Epidemiology & Public Health, University College Cork, Ireland, marie.reilly@ucc.ie
Agus Salim, Department of Statistics, University College Cork, Ireland, a.salim@ucc.ie

**Abstract:** The command `meanscor` that implements the mean score method of Reilly and Pepe (1995) for incorporating incomplete cases into logistic regression analysis through a weighted regression model is introduced and illustrated.

**Keywords:** missing data, mean score method, logistic regression.

## Background

Missing data is a common problem in statistical analysis. Perhaps the most popular approach when confronted with missing data is excluding the incomplete cases from analysis and proceeding to analyze the complete cases using standard methods. While valid under certain assumptions regarding the missingness mechanism, this approach results in a loss of precision due to the ignored observations. The mean score method of Reilly and Pepe (1995) allows us to incorporate the incomplete cases into logistic regression analysis through a weighted regression model. For random missingness, this results in an improvement in efficiency over the analysis of complete cases only. More importantly, the method is applicable to a wide range of patterns of missingness known as MAR (missing at random), where missingness may depend on the completely observed variables but not on the unobserved value of the incompletely observed variable(s).

## Syntax

`meanscor` *depvar* [ *indepvars* ] [`if` *exp*] [`in` *range*] [`, first(`*varlist*`) second(`*varlist*`) odd(`#`)` ]

## Description

The `meanscor` command performs a weighted logistic regression using the mean score method. This function requires the complete covariate(s) to be categorical, and the default output contains the regression coefficient estimates and their standard errors in odds-ratio form.

An important area of application of this function is in the analysis of data from a two-stage study. In this type of study, some variables are incomplete due to only a subset of the study subjects being sampled at the second stage (Reilly 1996).

## Options

`first(`*varlist*`)` specifies the complete covariates.

`second(`*varlist*`)` specifies the incomplete covariates.

`odd(#)` specifies whether the odds-ratio (`odd` $= 1$) or regression coefficients (`odd` $= 0$) format is reported. Default value is 1.

## Methods and Formulas

The mean score estimates will maximize the weighted likelihood

$$\sum_{i \in C} \left(1 + \frac{n^{I(z_i,y_i)}}{n^{C(z_i,y_i)}}\right) \log P_\beta(y_i|x_i)$$

where $n^{I(z_i,y_i)}$ is the number of incomplete observations in each stratum defined by the different levels of response $y_i$ and complete covariates $z_i$, and $n^{C(z_i,y_i)}$ is the number of complete observations in each stratum.

As the above equation indicates, the mean score method weights each complete observation according to the total number of observations in the same stratum.

The asymptotic variance of the mean score estimate is given by

$$\mathrm{Var}(\widehat{\beta}) = \frac{1}{n}\left(I^{-1} + I^{-1}VI^{-1}\right)$$

where $n$ is total number of observations, and $I$ is the usual information matrix. $V$ is estimated by the matrix

$$\sum_{(y,z)} \frac{n(y,z)n^{I(y,z)}}{n^{C(y,z)}} \mathrm{Var}(S_\beta|y,z)$$

where $\mathrm{Var}(S_\beta(y,z))$ is the variance–covariance matrix of the score in each $(y,z)$ stratum.

We can regard the second term of the variance expression as a penalty for the incompletely observed observations. Hence, the mean score estimates will have larger variance than the estimates obtained if all observations were complete but smaller variance than the estimates from an analysis of complete cases only.

## Examples

We begin with a simulated dataset. We generated 1,000 observations of a predictor variable $x$ from the standard normal distribution. The response variable $y$ was then generated as a Bernoulli random variable with $p = \exp(x)/\{1 + \exp(x)\}$. A dichotomous surrogate variable for $x$, called $z$, was generated as one for positive $x$ and zero otherwise.

A random subsample of 500 observations had their $x$ value deleted (set to missing). The dataset, called `sim_miss.dta` is provided with this insert as an illustration and can be analyzed using the mean score method by

```
. use sim_miss
. meanscor y x,first(z) second(x)
meanscore estimates
------------------------------------------------------------------------------
         |  odd-ratio   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
    cons |   1.050643   .0751759      0.690   0.490       .9131663    1.208817
       x |   2.770173    .282211     10.002   0.000       2.268772    3.382384
------------------------------------------------------------------------------
```

We can compare this to the logistic regression analysis using only the complete observations:

```
. keep if x~=.
. logit y x, or
Logit estimates                                   Number of obs   =        500
                                                  LR chi2(1)      =      92.97
                                                  Prob > chi2     =     0.0000
Log likelihood = -299.18928                       Pseudo R2       =     0.1345
------------------------------------------------------------------------------
         |  odd-ratio   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
       x |   2.771684   .3326964      8.493   0.000       2.190638    3.506847
------------------------------------------------------------------------------
```

Note that the mean score estimate above had smaller standard error, reflecting the additional information used in the analysis. Also, since $z$ is a surrogate for $x$, it is not used in the complete case analysis.

Next, we consider a real example of an application of the mean score method to a case–control study of the association between ectopic pregnancy and sexually transmitted diseases; see Reilly and Pepe (1995) for a full description of the data

```
. use ectopic
. meanscor y gonn-chlam,first(gonn-sexptn) second(chlam)
meanscore estimates
------------------------------------------------------------------------------
         |  odd-ratio   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
    cons |   .4543184   .0987123     -3.631   0.000       .2967666    .6955137
    gonn |   .9495978   .2856096     -0.172   0.863       .5266531    1.712201
   contr |   .0943838   .0176643    -12.612   0.000       .0654021    .1362082
  sexptn |   2.099286   .4938943      3.152   0.002       1.323766    3.329139
   chlam |   2.471606   .7808384      2.864   0.004       1.330653    4.590858
------------------------------------------------------------------------------
```

For comparison, an analysis of complete cases only gives

```
. keep if chlam ~=.
. logit y gonn-chlam, or
Logit estimates                                   Number of obs   =        327
                                                  LR chi2(4)      =     104.24
                                                  Prob > chi2     =     0.0000
Log likelihood = -169.54627                       Pseudo R2       =     0.2351
------------------------------------------------------------------------------
         |  odd-ratio   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
    gonn |   .7445515   .3132037     -0.701   0.483       .3264582    1.698095
   contr |   .1098308   .0303352     -7.997   0.000        .063918    .1887231
  sexptn |    1.93898   .7101447      1.808   0.071        .945853     3.97487
   chlam |    2.47682   .7576623      2.965   0.003       1.359912    4.511054
------------------------------------------------------------------------------
```

## References

Reilly, M. 1996. Optimal sampling strategies for two-stage studies. *American Journal of Epidemiology* 143: 92–100.

Reilly, M. and M. S. Pepe. 1995. A mean score method for missing and auxiliary covariate data in regression models. *Biometrika* 82: 299–314.

## sg157 | Predicted values calculated from linear or logistic regression models

Joanne M. Garrett, University of North Carolina, garrettj@med.unc.edu

**Abstract:** The program `predcalc` for easily calculating predicted values and confidence intervals from linear or logistic regression model estimates for specified values of the $X$ variables is introduced and illustrated.

**Keywords:** regression models, predicted values.

## Syntax

`predcalc` *yvar*, `xvar(`*xvarlist*`)` [ `level(#)` `model` `linear` ]

## Description

`predcalc` ("prediction calculator") is an easy method of calculating predicted values and confidence intervals from linear or logistic regression model estimates for specified values of the $X$ variables. If no model has been fit previously (using `regress` or `logistic`), `predcalc` will set up and run the model based on the $Y$ variable *yvar* (continuous or binary) and the specified $X$'s.

## Options

`xvar(`*xvarlist*`)` lists the $X$ variables and their values to use to solve the model equation. The model is based only on the $X$'s listed in this option. For example, `xvar(age=40 gender=1 chl=250)` specifies that the equation for a model including the variables `age`, `gender`, and `chl` (cholesterol) will be solved using the values of 40 years old, male, and a cholesterol level of 250.

`level(#)` specifies the confidence level, in percent, for confidence intervals for predicted values. The default is `level(95)` or as set by `set level`.

`model` displays the regression table. This option is not needed to estimate the model. It is simply for display purposes.

`linear` causes a model with a binary outcome to be fit using linear, rather than logistic regression. This is a rarely used option.

## Example 1

All the examples come from a cohort study of coronary heart disease (Cassel 1971). The data consist of 609 men who are followed for seven years to see what variables are risk factors for an elevated systolic blood pressure `sbp` (measured in mmHg) or for coronary heart disease `chd` (1 for yes, 0 for no). Some of the $X$ variables are serum catecholamine level `cat` (1 for high, 0 for low), smoking status `smk` (1 for current smoker, 0 for nonsmoker), regular exercise `exer` (1 for yes, 0 for no), the men's age `age` in years, and cholesterol level `chl` in mg per 100 ml.

```
. use chd
. describe
Contains data from chd.dta
  obs:           609                          Evans County Data
 vars:             7                          2 Mar 2000 20:45
 size:         7,917 (99.1% of memory free)
-------------------------------------------------------------------------------
   1. sbp       int    %8.0g                 Systolic blood pressure
   2. chd       byte   %8.0g       hilo      Coronary heart disease
   3. cat       byte   %8.0g       hilo      Serum catecholamine level
   4. smk       byte   %8.0g       yesno     Smoking status
   5. exer      byte   %8.0g       yesno     Regular exercise
   6. age       byte   %8.0g                 Age in years
   7. chl       int    %8.0g                 Serum cholesterol
-------------------------------------------------------------------------------
```

After fitting a linear regression model we find that older age, high catecholamine, no exercise, and higher cholesterol are significant predictors of higher systolic blood pressure.

```
. reg sbp age cat exer chl
  Source |       SS       df       MS          Number of obs =     609
---------+------------------------------       F(  4,   604) =   85.91
   Model |  166731.045     4  41682.7613       Prob > F      =  0.0000
Residual |   293038.86   604  485.163675       R-squared     =  0.3626
---------+------------------------------       Adj R-squared =  0.3584
   Total |  459769.905   608  756.200501       Root MSE      =  22.026
------------------------------------------------------------------------------
   sbp |      Coef.   Std. Err.      t    P>|t|   [95% Conf. Interval]
-------+----------------------------------------------------------------
   age |   .2724588   .1054554    2.584   0.010   .0653549    .4795627
   cat |   33.99951   2.622221   12.966   0.000   28.84973    39.14929
  exer |  -8.783763   2.173233   -4.042   0.000  -13.05177   -4.515753
   chl |   .0741253   .0227488    3.258   0.001    .029449    .1188015
 _cons |   114.7267   7.441563   15.417   0.000   100.1123    129.3412
------------------------------------------------------------------------------
```

Given this model, suppose we would like to see the predicted systolic blood pressure for a 60 year old with high catecholamine (`cat` $= 1$), no regular exercise (`exer` $= 0$), and a cholesterol level of 260. Because `sbp` is continuous, `predcalc` defaults to linear regression and solves the equation using the elements of $\beta$ multiplied by the specified $X$ values. Also, it calculates a

confidence interval around the predicted value based on the standard error of the prediction. Each variable in the model is set to the desired value using the `xvar` option.

```
. predcalc sbp, xvar(age=60 cat=1 exer=0 chl=260)
Model:     Linear Regression
Outcome:   Systolic blood pressure -- sbp
X Values:  age=60 cat=1 exer=0 chl=260
Num. Obs:  609
Predicted Value and 95% CI for sbp:
     184.35  ( 179.31,  189.38)
```

The predicted value for systolic blood pressure is 184.35 with a 95% confidence interval of 179.3 to 189.4. Had we not run the model previously, `predcalc` would still work. The command first looks for stored estimates, and if they are not found, the appropriate model is run. The model is not shown unless requested with the `model` option. In either case, it is a good idea to check the "$X$ Values" list to make sure that the predicted estimate is based on the model and variables expected, since the model will contain only on the $X$'s listed in the `xvar` option.

## Example 2

Next we will use the same model but change some of the values for the $X$ variables. This time, we will request the predicted systolic blood pressure for a 40 year old with low catecholamine (`cat` $= 0$), who exercises regularly (`exer` $= 1$), and has a cholesterol level of 200.

```
. predcalc sbp, xvar(age=40 cat=0 exer=1 chl=200)
Model:     Linear Regression
Outcome:   Systolic blood pressure -- sbp
X Values:  age=40 cat=0 exer=1 chl=200
Num. Obs:  609
Predicted Value and 95% CI for sbp:
     131.67  ( 128.42,  134.91)
```

This predicted value for systolic blood pressure (131.67) is quite a bit lower than the previous example for an individual with stronger risk factors for hypertension.

## Example 3

Rather than using systolic blood pressure as the outcome, we will look at the dichotomous variable `chd` for coronary heart disease (1 for yes, 0 for no). We can use logistic regression, but instead of running the model first, we can use `predcalc`. Suppose we want to know the probability of coronary heart disease for a person with strong risk factors: 60 years old, smokes, does not exercise, and has a cholesterol value of 260. Because `chd` is binary, a logistic regression model is assumed and run. The `model` option prints a copy of the model. (Remember, `model` is optional and is not needed to run the model. It just displays the regression table of estimates used to solve the equation).

```
. predcalc chd, xvar(age=60 smk=1 exer=0 chl=260) model
Logit estimates                           Number of obs   =        609
                                          LR chi2(4)      =      30.55
                                          Prob > chi2     =     0.0000
Log likelihood = -204.00576               Pseudo R2       =     0.0697
------------------------------------------------------------------------------
         chd | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   1.046986   .0143708     3.345   0.001    1.019195    1.075534
         smk |   2.408027   .7311962     2.894   0.004    1.327989    4.366448
        exer |    .532516   .1453497    -2.309   0.021    .3118876    .9092162
         chl |   1.007934   .0031807     2.504   0.012     1.00172    1.014188
------------------------------------------------------------------------------
Model:     Logistic Regression
Outcome:   Coronary heart disease -- chd
X Values:  age=60 smk=1 exer=0 chl=260
Num. Obs:  609
Predicted Value and 95% CI for chd:
    0.3177  (0.2141, 0.4432)
```

The probability of developing coronary heart disease for someone with these attributes is 0.32 with 95% confidence interval from 0.21 to 0.44.

### Example 4

Using the same model, we request the probability of coronary heart disease for a 40 year old who does not smoke, exercises regularly, and has a cholesterol value of 200.

```
. predcalc chd, xvar(age=40 smk=0 exer=1 chl=200)
Model:     Logistic Regression
Outcome:   Coronary heart disease -- chd
X Values:  age=40 smk=0 exer=1 chl=200
Num. Obs:  609
Predicted Value and 95% CI for chd:
    0.0249  (0.0118, 0.0518)
```

For the individual with no strong risk factors, the probability of developing coronary heart disease is only 0.02.

### References

Cassel, J. C. 1971. Summary of major findings of the Evans County heart disease study. *Archives of Internal Medicine* 128(8): 887–889.

## snp15.2 Update to Somersd

Roger Newson, Guy's, King's and St Thomas' School of Medicine, London, UK, roger.newson@kcl.ac.uk

**Abstract:** `somersd` calculates confidence intervals for rank order statistics. It has been updated to handle long variable lists.

**Keywords:** Somers' D, Kendall's tau, rank correlation, confidence intervals, nonparametric methods.

The command `somersd` introduced in Newson (2000a) and updated in Newson (2000b) has again been updated, this time to handle long variable lists (it was previously limited to lists of 8 variables). It has also been improved, streamlined, debugged, and intensively certified.

### References

Newson, R. 2000a. snp15: `somersd` - Confidence limits for nonparametric statistics and their differences. *Stata Technical Bulletin* 55: 47–55.

——. 2000b. snp15.1: Update to somersd. *Stata Technical Bulletin* 57: 35.

## snp16 Robust confidence intervals for median and other percentile differences between two groups

Roger Newson, Guy's, King's and St Thomas' School of Medicine, London, UK, roger.newson@kcl.ac.uk

**Abstract:** A program is presented for calculating robust confidence intervals for median (and other percentile) differences (and ratios) between values of a variable in two samples. The median difference is the same as that produced by the program `cid`, using the Conover method. However, the confidence limits are typically different, being robust to the possibility that the two population distributions differ in ways other than location, such as having unequal variances. The program uses `somersd`.

**Keywords:** robust, confidence interval, median, percentile, difference, ratio, rank-sum, Wilcoxon, two-sample.

### Syntax

`cendif` *depvar* [`using` *filename*] [*weight*] [`if` *exp*] [`in` *range*], `by(`*groupvar*`)` [ `centile(`*numlist*`)`

`level(#)` `eform` `cluster(`*varname*`)` `tdist` `transf(`{`z` | `asin` | `iden`}`)`

`saving(`*filename*[`,replace`]`)` `nohold` ]

`fweights`, `iweights` and `pweights` are allowed. They are treated as described in *Methods and formulas* below.

### Description

`cendif` calculates confidence intervals for median differences, and other percentile differences, between values of a $Y$-variable in *depvar* for a pair of observations chosen at random from two groups $A$ and $B$, defined by the *groupvar* in the `by` option. These confidence intervals are robust to the possibility that the population distributions in the two groups are different in ways other than location. This might happen if, for example, the two populations had different variances. For positive-valued variables, `cendif` can be used to calculate confidence intervals for median ratios or other percentile ratios. `cendif` requires the program `somersd` from Newson (2000).

## Options

`by(`*groupvar*`)` is not optional. It specifies the name of the grouping variable. This variable must have exactly two possible values. The lower value indicates Group $A$, and the higher value indicates Group $B$.

`centile(`*numlist*`)` specifies a list of percentile differences to be reported and defaults to `centile(50)` (median only). Specifying `centile(25 50 75)` will produce the 25th, 50th, and 75th percentile differences.

`level(#)` specifies the confidence level, in percent, for confidence intervals. The default is `level(95)` or as set by `set level`.

`eform` specifies that exponentiated percentile differences are to be given. This option is used if *depvar* is the log of a positive-valued variable. In this case, confidence intervals are calculated for percentile ratios between values of the original positive variable, instead of for percentile differences.

`cluster(`*varname*`)` specifies the variable which defines sampling clusters. If `cluster` is defined, then the percentiles are calculated using the between-cluster Somers' $D$, and the confidence intervals are calculated assuming that the data are a sample of clusters from a population of clusters, rather than a sample of observations from a population of observations.

`tdist` specifies that the standardized Somers' $D$ estimates are assumed to be sampled from a $t$ distribution with $n-1$ degrees of freedom, where $n$ is the number of clusters or the number of observations if `cluster` is not specified.

`transf(`*transformation_name*`)` specifies that the Somers' $D$ estimates are to be transformed, defining a standard error for the transformed population value from which the confidence limits for the percentile differences are calculated. `z` (the default) specifies Fisher's $z$ (the hyperbolic arctangent), `asin` specifies Daniels' arcsine, and `iden` specifies identity or untransformed.

`saving(`*filename*`[,replace])` specifies a dataset, to be created, whose observations correspond to the observed values of differences between a value of *depvar* in Group A and a value of *depvar* in Group B. `replace` instructs Stata to replace any existing dataset of the same name. The saved dataset can then be reused if `cendif` is called later, with `using`, to save the large amounts of processing time used to calculate the set of observed differences. The `saving` option and the `using` utility are provided mainly for programmers to use, at their own risk.

`nohold` indicates that any existing estimation results are to be overwritten with a new set of estimation results, for the use of programmers. By default, any existing estimation results are restored after execution of `cendif`.

## Remarks

`cendif` calls `somersd` (see Newson 2000), which has been updated, in order to take long variable lists. (It was previously limited to eight variables.)

## Methods and formulas

Suppose that a population contains two disjoint subpopulations $A$ and $B$, and a random variable $Y$ is defined for individuals from both subpopulations. For $0 < q < 1$, a $100q$th percentile difference in $Y$ between Populations $A$ and $B$ is defined as a value $\theta$ satisfying

$$D[Y^*(\theta)|X] = 1 - 2q \tag{1}$$

where $X$ is a binary variable equal to 1 for Population $A$ and 0 for Population $B$, $Y^*(\theta)$ is defined as $Y$ if $X = 1$ and $Y + \theta$ if $X = 0$, and $D[\,\cdot\,|\,\cdot\,]$ denotes Somers' $D$ (Somers 1962, Newson 2000). Somers' $D$ is defined as

$$D[V|W] = E\left[\operatorname{sign}(V_1 - V_2)\operatorname{sign}(W_1 - W_2)\right] / E\left[\operatorname{sign}(W_1 - W_2)^2\right] \tag{2}$$

where $(W_1, V_1)$ and $(W_2, V_2)$ are bivariate data points sampled independently from the same population, and $E[\,\cdot\,]$ denotes expectation. In the case of (1), where $W = X$ and $V = Y^*(\theta)$, Somers' $D$ is the difference between two conditional probabilities. Given an individual sampled from Population $A$ and an individual sampled from Population $B$, these are the probability that the individual from Population $A$ has the higher $Y^*$ value and the probability that the individual from Population $B$ has the higher $Y^*$ value. Somers' $D$ is therefore the parameter equal to zero under the null hypothesis tested by the "nonparametric" Wilcoxon rank-sum test on $Y^*(\theta)$. In the case where $q = 0.5$ (and therefore $1 - 2q = 0$), a $100q$th percentile difference is known as a median percentile difference and is zero under the null hypothesis tested by a Wilcoxon rank-sum test on $Y$.

Note that a value of $\theta$ satisfying (1) is not always unique. If $Y$ has a discrete distribution, then there may be no solution or a wide interval of solutions. However, the method used here is intended to produce a confidence interval containing any given $\theta$ satisfying (1), with a probability at least equal to the confidence level, if such a $\theta$ exists.

We will assume that there are $N_1$ observations sampled from Population $A$ and $N_2$ observations sampled from Population $B$, giving a total of $N_1 + N_2 = N$ observations. These observations will be identified by double subscripts, so that $Y_{ij}$ is the $Y$ value for the $j$th observation sampled from the $i$th population (where $i = 1$ for Population $A$ and $i = 2$ for Population $B$). The corresponding $X$ values (ones and zeros) will be denoted $X_{ij}$. The observations will be assumed to have importance weights

(`iweights` or `pweights`) denoted by $w_{ij}$ and cluster sequence numbers denoted by $c_{ij}$. `cendif` follows the usual Stata practice of assuming an `fweight` to stand for multiple observations with the same values for all other variables. The clusters may be nested within the two groups or contain observations from each of the two groups, but the percentile differences will only apply to observations from distinct clusters. If clusters are present, then the confidence intervals will be calculated assuming that the sample was generated by sampling clusters independently from a population of clusters, rather than by sampling $N$ observations independently from the total population of observations or by sampling $N_1$ and $N_2$ observations from Populations $A$ and $B$, respectively. (By default, all the $w_{ij}$ will be ones, and the $c_{ij}$ will be in sequence from 1 to $N$. The difference between these three alternatives will not matter.) We will denote by $M$ the number of distinct values of a difference, $Y_{1j} - Y_{2k}$, observed between $Y$ values in the two samples belonging to different clusters. The difference values themselves will be denoted by $t_1, \ldots, t_M$. For each $h$ from 1 to $M$, we define the sum of product weights of differences equal to $t_h$ as

$$W_h = \sum_{j,k:\, Y_{1j} - Y_{2k} = t_h} \delta(c_j, c_k) w_{1j} w_{2k} \tag{3}$$

where $\delta(a, b)$ is 0 if $a = b$ and 1 if $a \neq b$. Given a value of $\theta$ expressed in units of $Y$, we can define $Y^*_{ij}(\theta)$ to be $Y_{ij}$ if $i = 1$ and $Y_{ij} + \theta$ if $i = 2$. The sample Somers' $D$ of $Y^*(\theta)$ with respect to $X$ is defined as

$$\begin{aligned} D^*(\theta) = \widehat{D}[Y^*(\theta)|X] &= \frac{\sum_{j=1}^{N_1} \sum_{k=1}^{N_2} \delta(c_{1j}, c_{2k})\, w_{1j} w_{2k} \operatorname{sign}(Y_{1j} - Y_{2k} - \theta)}{\sum_{j=1}^{N_1} \sum_{k=1}^{N_2} \delta(c_{1j}, c_{2k})\, w_{1j} w_{2k}} \\ &= \frac{\sum_{h:t_h > \theta} W_h - \sum_{h:t_h < \theta} W_h}{\sum_{h=1}^{M} W_h} \end{aligned} \tag{4}$$

where $\widehat{D}[\,\cdot\,|\,\cdot\,]$ denotes the sample Somers' $D$, defined by the methods of Newson (2000). Clearly, given a sample, $D^*(\theta)$ is a nonincreasing function of $\theta$. (Note that only between-cluster differences are included.) Figure 1 shows $D^*(\theta)$ as a function of $\theta$ for differences between trunk capacities of American and foreign cars (expressed in cubic feet) in the `auto` data. The squares represent the values $D^*(t_h)$ for the observed differences $t_h$. Note that $D^*(\theta)$ is discontinuous at the observed differences, and constant in each open interval between two successive observed differences.

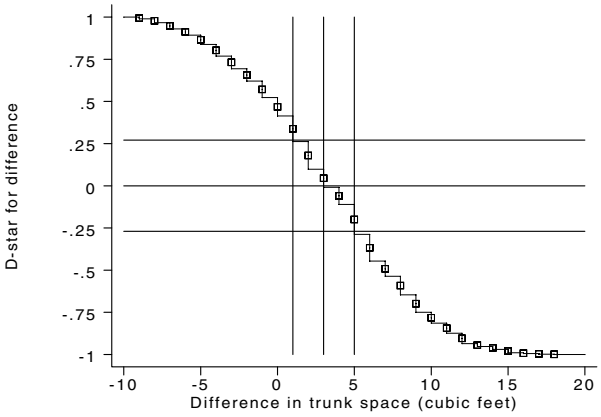



Figure 1. $D^*(\theta)$ plotted against the difference $\theta$ in trunk space between American and foreign cars

We aim to include $\theta$ in a confidence interval for a $q$th percentile difference if, and only if, the sample $D^*(\theta)$ is compatible with a *population* $D[Y^*(\theta)|X]$ equal to $1 - 2q$. The methods of Newson (2000), used by the program `somersd`, typically use a transformation $\zeta(\,\cdot\,)$, which, for present purposes, may either be the identity, the arcsine or Fishers' $z$ (the hyperbolic arctangent). The transformed sample statistic $\widehat{\zeta}(\theta) = \zeta[D^*(\theta)]$ is assumed to be normally distributed around the population parameter $\zeta\{D[Y^*(\theta)|X]\}$. In the present application, we assume that if $D[Y^*(\theta)|X] = 1 - 2q$, then the quantity

$$[\,\widehat{\zeta}(\theta) - \zeta(1 - 2q)\,] \,/\, \mathrm{SE}[\widehat{\zeta}(\theta)] \tag{5}$$

has a standard Normal distribution, where $\mathrm{SE}[\widehat{\zeta}(\theta)]$ is the sampling standard deviation (or standard error) of $\zeta[D^*(\theta)]$. If we knew the value of $\mathrm{SE}[\widehat{\zeta}(\theta)]$, then a $100(1 - \alpha)\%$ confidence interval for a $q$th percentile difference might be the interval of values of $\theta$ for which

$$\zeta^{-1}\{\,\zeta(1 - 2q) - z_\alpha \,\mathrm{SE}[\widehat{\zeta}(\theta)]\,\} \leq D^*(\theta) \leq \zeta^{-1}\{\,\zeta(1 - 2q) + z_\alpha \,\mathrm{SE}[\widehat{\zeta}(\theta)]\,\} \tag{6}$$

where $z_\alpha$ is the $100(1-\alpha/2)$th percentile of the standard Normal distribution.

To construct such a confidence interval, we proceed as follows. Given a value of $D$, define

$$B_{\rm L}(D) = \inf\{\theta : D^*(\theta) \le D\}, \quad B_{\rm R}(D) = \sup\{\theta : D^*(\theta) \ge D\},$$

$$B_{\rm C}(D) = \begin{cases} B_{\rm L}(D), & \text{if } B_{\rm R}(D) = \infty \\ B_{\rm R}(D), & \text{if } B_{\rm L}(D) = -\infty \\ [\,B_{\rm L}(D) + B_{\rm R}(D)\,]/2, & \text{otherwise} \end{cases} \tag{7}$$

(By convention, the supremum (or infimum) of a set unbounded to the right (or left) are defined as $\infty$ and $-\infty$, respectively.) Clearly, $B_{\rm L}(D) \le B_{\rm C}(D) \le B_{\rm R}(D)$, and the values of $B_{\rm L}(D)$ and $B_{\rm R}(D)$ (if finite) can be either the same $t_h$ or two successive ones. The confidence interval for the $q$th percentile difference is centered on the sample $q$th percentile difference

$$\widehat{\xi}_q = B_{\rm C}(1-2q) \tag{8}$$

`cendif` then calls `somersd`, with the $X_{ij}$ as the predictor variable, and the $Y^*_{ij}(\widehat{\xi}_q)$, for the values of $q$ implied by the `centile` option, as the predicted variables. The standard errors generated by `somersd` are used as estimates $\widehat{\rm SE}[\widehat{\zeta}(\widehat{\xi}_q)]$ of the standard error of $\widehat{\zeta}(\theta)$ where $\theta$ satisfies (1). The lower and upper confidence limits for the $q$th percentile difference are, respectively,

$$\widehat{\xi}^{(\min)}_q = B_{\rm L}\Big(\zeta^{-1}\big\{\,\zeta(1-2q) - z_\alpha\,\widehat{\rm SE}[\widehat{\zeta}(\widehat{\xi}_q)]\,\big\}\Big), \quad \widehat{\xi}^{(\max)}_q = B_{\rm R}\Big(\zeta^{-1}\big\{\,\zeta(1-2q) + z_\alpha\,\widehat{\rm SE}[\widehat{\zeta}(\widehat{\xi}_q)]\,\big\}\Big) \tag{9}$$

If `tdist` is specified, then `cendif` uses the $t$ distribution with $N-1$ degrees of freedom (or $N_{\rm clust}-1$ degrees of freedom if there are $N_{\rm clust}$ clusters) instead of the normal distribution, so $t_\alpha$ replaces $z_\alpha$ in (6) and (9). Note that the upper and lower confidence limits may occasionally be infinite, in the case of extreme percentiles and/or very small sample numbers. (`cendif` codes these infinite limits as plus or minus the "magic number" `1E+300`, or $\pm 10^{300}$.) Figure 1 shows the median difference in trunk capacity, and its confidence limits, as reference lines on the horizontal axis. The estimated median difference is 3 cubic feet, with 95% confidence limits from 1 to 5 cubic feet. The reference lines on the vertical axis are the optimum, minimum, and maximum values of $D^*(\theta)$ required for $\theta$ to be in the confidence interval. These values of $D^*(\theta)$ are saved by `cendif` in the matrix `r(Dsmat)`. If the option `saving` is specified, then `cendif` also saves an output dataset with $M$ observations corresponding to the ordered differences $t_h$. The variables are `diff` (containing the $t_h$), `weight` (containing the $W_h$), `Dstar` (containing the $D^*(t_h)$), and `Dstar_r`, which contains the right-hand limiting value of $D^*(\theta)$,

$$D^*_R(t_h) = \lim_{\theta \to t_h +} D^*(\theta) \tag{10}$$

which is the value of $D^*(\theta)$ in the open interval $(t_h, t_{h+1})$ for $h < M$. Conover (1980) presents a method which, for large samples, is essentially equivalent to (6), in the special case where $q = 0.5$ and $\zeta(D) = D$. (This is the method for calculating confidence intervals for median differences popularized by Campbell and Gardner (1988) and Gardner and Altman (1989), and available in Stata using Patrick Royston's `cid` routine, currently on the Ideas list (Royston 1998).) However, Conover's method uses the assumption that the two population distributions are different only in location. This assumption (essentially) enables the calculation of ${\rm SE}[\widehat{\zeta}(\theta)]$ for large samples and the exact distribution of $D^*(\theta)$ for small samples. It also implies that the median difference is the difference between medians. In the present case, we are not making this assumption, as the confidence interval is intended to be robust to the possibility that the two populations are different in ways other than location. (For instance, the two populations might be unequally variable.) The median difference is therefore not necessarily the difference between medians. Also, we have to estimate ${\rm SE}[\widehat{\zeta}(\theta)]$, and this estimate is itself subject to some amount of sampling error. The method of `cendif` compares to Conover's method as the unequal-variance $t$-test compares to the equal-variance $t$-test. Conover's method, like the equal-variance $t$-test, assumes that you can use data from the larger of two samples to estimate the population variability of the smaller sample.

I have been carrying out some simulations of sampling from two normal populations, with a view to finding the coverage probabilities and geometric mean lengths of the confidence intervals for the median difference generated by `cid` and by `cendif` with the `tdist` option. So far, I find that, even with small sample sizes, the `cendif` method consistently gives coverage probabilities closer to the nominal value than the Conover method when variances are unequal, in which case `cid` produces confidence intervals either too wide or too narrow, depending on whether the larger or smaller sample has the greater population variance. Usually, the difference in coverage probability is small (1% or 2%), so the Conover method performs fairly well, in spite of false assumptions. However, if a sample of 20 is compared to a sample of 10, and the population standard deviation of the smaller sample is three times that of the larger sample, then the nominal 95% confidence interval has a true coverage

probability of only 90% under the Conover method, compared to 94% under the `cendif` method. The two methods show little or no difference, either in geometric mean confidence interval width or in coverage probability, when the variances are equal and the Conover assumption is therefore true. From the results so far, I would recommend the `cendif` method as an improved version of the Conover method, offering insurance against the possibility that the Conover assumption is wildly wrong, at little or no price in performance if the Conover assumption is right. However, I am planning to carry out further simulations on the two methods and to report the results in due course.

### Example 1

In the `auto` data, we compare weights of American and foreign cars. We use `cid` and `cendif` to estimate the median difference:

```
. cid weight,by(foreign) median unpaired
Rank-based confidence interval for difference in  medians by foreign
Variable |     Obs     Estimate           K        [95% Conf. Interval]
---------+------------------------------------------------------------
  weight |      74         1095         406             720        1350
. cendif weight,by(foreign)
Y-variable: weight (Weight (lbs.))
Grouped by: foreign (Car type)
Group numbers:
   Car type |      Freq.     Percent        Cum.
------------+-----------------------------------
   Domestic |         52       70.27       70.27
    Foreign |         22       29.73      100.00
------------+-----------------------------------
      Total |         74      100.00
Transformation: Fisher´s z
95% confidence interval(s) for percentile difference(s)
between values of weight in first and second groups:
     Percent  Pctl_Dif   Minimum   Maximum
r1        50      1095       750      1330
```

We note that the median difference in weight is 1,095 pounds according to both `cid` and `cendif`. However, the confidence limits given by `cendif` are 750 and 1,330 pounds, whereas the confidence limits given by `cid` are 720 and 1,350 pounds. This is because foreign cars are fewer in number and less variable in weight than American cars, and `cid` assumes equal variances, whereas `cendif` allows for unequal variances. If we carry out equal-variance and unequal-variance $t$ tests (not shown), we find a similar difference in the width of the confidence limits for the mean difference.

`cendif` can also calculate confidence intervals for percentiles other than medians. These contain information about the degree of overlap between the two populations. Here, we estimate the 25th, 50th, and 75th percentile differences, using the `centile` option.

```
. cendif weight,by(foreign) ce(25 50 75)
Y-variable: weight (Weight (lbs.))
Grouped by: foreign (Car type)
Group numbers:
   Car type |      Freq.     Percent        Cum.
------------+-----------------------------------
   Domestic |         52       70.27       70.27
    Foreign |         22       29.73      100.00
------------+-----------------------------------
      Total |         74      100.00
Transformation: Fisher´s z
95% confidence interval(s) for percentile difference(s)
between values of weight in first and second groups:
     Percent  Pctl_Dif   Minimum   Maximum
r1        25       485       100       810
r2        50      1095       750      1330
r3        75      1555      1320      1790
```

If we want to estimate percentile ratios of weight, rather than percentile differences, then we simply take logs and use the `eform` option.

```
. gene logwt=log(weight)
. cendif logwt,by(foreign) ce(25 50 75) eform
```

```
Y-variable: logwt
Grouped by: foreign (Car type)
Group numbers:
   Car type |      Freq.     Percent        Cum.
------------+-----------------------------------
   Domestic |         52       70.27       70.27
    Foreign |         22       29.73      100.00
------------+-----------------------------------
      Total |         74      100.00
Transformation: Fisher´s z
95% confidence interval(s) for percentile ratio(s)
between values of exp(logwt) in first and second groups:
      Percent   Pctl_Rat    Minimum    Maximum
r1         25  1.1935375  1.0341465  1.3533567
r2         50  1.4806389  1.3101849  1.6280196
r3         75   1.744916  1.6079542  1.8772724
```

We note that, typically, American cars are 148% heavier than foreign cars, with confidence limits ranging from 131% to 163% as heavy. The 25th percentile ratio (103% to 135%) shows that the two car types do not overlap a great deal.

## Saved results

`cendif` saves in `r()`:

Scalars

| | | | |
|---|---|---|---|
| `r(N)` | number of observations | `r(N_clust)` | number of clusters |
| `r(N_1)` | sample size $N_1$ | `r(N_2)` | sample size $N_2$ |
| `r(df_r)` | residual degrees of freedom (if `tdist` present) | | |

Macros

| | | | |
|---|---|---|---|
| `r(depvar)` | name of dependent variable | `r(by)` | name of `by` variable defining groups |
| `r(clustvar)` | name of cluster variable | `r(tdist)` | `tdist` if specified |
| `r(wtype)` | weight type | `r(wexp)` | weight expression |
| `r(centiles)` | list of percents for percentiles | `r(Dslist)` | list of $D^*$-values for percentiles |
| `r(transf)` | transformation specified by `transf` | `r(tranlab)` | transformation label in output |
| `r(level)` | confidence level | `r(eform)` | `eform` if specified |

Matrices

| | | | |
|---|---|---|---|
| `r(cimat)` | confidence intervals for differences or ratios | `r(Dsmat)` | upper and lower limits for $D^*(\theta)$ |

## Acknowledgments

I would like to thank Nick Cox of Durham University, UK, and Bill Gould of Stata Corporation for some very helpful advice on the coding of infinite confidence limits, such as those occasionally resulting from Equation (9).

## References

Campbell, M. J. and M. J. Gardner. 1988. Calculating confidence intervals for some non-parametric analyses. *British Medical Journal* 296: 1454–1456.

Conover, W. J. 1980. *Practical Nonparametric Statistics*. 2d ed. New York: John Wiley & Sons.

Gardner, M. J. and D. G. Altman. 1989. *Statistics with Confidence*. London: British Medical Journal.

Newson, R. 2000. snp15.2: Update to somersd. *Stata Technical Bulletin* 58: 30.

Royston, P. 1998. CID: Stata module to calculate confidence intervals for means or differences. On the Ideas list at http://ideas.uqam.ca/ideas/data/Softwares/bocbocodeS338001.html.

Somers, R. H. 1962. A New asymmetric measure of association for ordinal variables. *American Sociological Review* 27: 799–811.

## sts15.1 | Tests for stationarity of a time series: update

Christopher F. Baum, Boston College, baum@bc.edu
Richard Sperling, The Ohio State University, rsperling@boo.net

**Abstract:** Enhances the Elliott–Rothenberg-Stock DF-GLS test and the Kwiatkowski–Phillips–Schmidt–Shin KPSS tests for stationarity of a time series introduced in Baum (2000) and corrects an error in both routines.

**Keywords:** stationarity, unit root, time series.

## Changes to dfgls

`dfgls` did not handle missing initial values properly. That is, if the time series variable specified had initial values not excluded by `if` or `in` conditions, those values were improperly considered in the construction of the sample size. This would apply as well to the consideration of variables with time series operators, such as `D.gdp`, since those variables will have at least one missing observation at the outset. This has been corrected.

The `dfgls` routine has been enhanced to add a very powerful lag selection criterion, the "modified AIC" (MAIC) criterion proposed by Ng and Perron (2000). They have established that use of this MAIC criterion may provide "huge size improvements" in the `dfgls` test. The criterion, indicating the appropriate lag order, is printed on `dfgls` output and may be used to select the test statistic from which inference is to be drawn.

It should be noted that all of the lag length criteria employed by `dfgls` (the sequential *t* test of Ng and Perron 1995, the SC, and the MAIC) are calculated, for various lags, by holding the sample size fixed at that defined for the longest lag. These criteria cannot be meaningfully compared over lag lengths if the underlying sample is altered to use all available observations. That said, if the optimal lag length (by whatever criterion) is found to be much less than that picked by the Schwert criterion, it would be advisable to rerun the test with the `maxlag` option specifying that optimal lag length, especially when using samples of modest size.

## New syntax for kpss

`kpss` *varname* [`if` *exp*] [`in` *range*] [`, maxlag(#) notrend qs auto`]

`kpss` did not make use of all available observations in the computation of the autocovariance function. This has been corrected. The online help file now provides instructions for reproducing the statistics reported in Kwiatkowski et al. (1992) from a dataset available online.

The `kpss` routine has been enhanced to add two options recommended by the work of Hobijn et al. (1998). An automatic bandwidth selection routine has been added, rendering it unnecessary to evaluate a range of test statistics for various lags. An option to weight the empirical autocovariance function by the quadratic spectral kernel, rather than the Bartlett kernel employed by KPSS, has also been introduced. These options may be used separately or in combination. It is in combination that Hobijn et al. found the greatest improvement in the test: "Our Monte Carlo simulations show that the best small sample results of the test in case the process exhibits a high degree of persistence are obtained using both the automatic bandwidth selection procedure and the Quadratic Spectral kernel" (1998, 14).

## New options

`qs` specifies that the autocovariance function is to be weighted by the quadratic spectral kernel, rather than the Bartlett kernel. Andrews (1991) and Newey and West (1994) "indicate that it yields more accurate estimates of $\sigma_\epsilon^2$ than other kernels in finite samples" (Hobijn et al. 1998, 6).

`auto` specifies that the automatic bandwidth selection procedure proposed by Newey and West (1994), as described by Hobijn et al. (1998, 7), is used to determine `maxlag` in two stages. First, the "a priori nonstochastic bandwidth parameter" $n_T$ is chosen as a function of the sample size and the specified kernel. The autocovariance function of the estimated residuals is calculated and used to generate $\gamma$ as a function of sums of autocorrelations. The `maxlag` to be used in computing the long-run variance, $\widehat{m}_T$, is then calculated as $\min\left[T, \text{int}\left[\widehat{\gamma}T^{\theta}\right]\right]$ where $\theta = 1/3$ for the Bartlett kernel and $\theta = 1/5$ for the quadratic spectral kernel.

## Additional saved results

`dfgls` saves the modified AIC at lag $n$ in `r(maic`*n*`)`.

## References

Andrews, D. W. K. 1991. Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica* 59: 817–858.

Baum, C. F. 2000. sts15: Test for stationarity of a time series. *Stata Technical Bulletin* 57: 36–39.

Hobijn, B., P. H. Franses, and M. Ooms. 1998. Generalizations of the KPSS-test for stationarity. Econometric Institute Report 9802/A, Econometric Institute, Erasmus University Rotterdam. `http://www.eur.nl/few/ei/papers`.

Kwiatkowski, D., P. C. B. Phillips, P. Schmidt, and Y. Shin. 1992. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of Econometrics* 54: 159–178.

Newey, W. K. and K. D. West. 1994. Automatic lag selection in covariance matrix estimation. *Review of Economic Studies* 61: 631–653.

Ng, S. and P. Perron. 1995. Unit root tests in ARMA models with data-dependent methods for the selection of the truncation lag. *Journal of the American Statistical Association* 90: 268–281.

——. 2000. Lag length selection and the construction of unit root tests with good size and power. *Econometrica*, in press.

| sxd2 | Computing optimal sampling designs for two-stage studies |
|---|---|

Marie Reilly, Epidemiology & Public Health, University College Cork, Ireland, marie.reilly@ucc.ie
Agus Salim, Department of Statistics, University College Cork, Ireland, a.salim@ucc.ie

**Abstract:** Commands are given for determining optimal sampling designs subject to fixed sample size, fixed budget, and fixed precision, and each command illustrated by an example.

**Keywords:** two-stage studies, mean score method.

## Background

The commands supplied here apply to two-stage studies where a dichotomous outcome variable $y$ and some categorical covariate(s) $z$ are available for all study subjects at the first stage. While at the second stage, a subset of the study subjects have some additional covariate(s) $x$ measured. The second-stage covariates may be continuous. The second stage subjects can be a stratified random sample, where the strata are defined by the levels of $y$ and $z$. The mean score algorithm (Reilly and Pepe 1995) allows us to analyze the data from such a two-stage study incorporating all first and second stage observations.

The variance expression of the mean score estimate given by Reilly and Pepe (1995) shows that the variance depends on 1) the total number of observations and 2) the second-stage sampling fractions in each of the strata defined by the different levels of response $y$ and first stage covariates $z$. Thus it is possible to minimize the variance of a particular variable by optimally choosing the number of observations and/or the second-stage sampling fractions.

## Syntax

`optfixn` *depvar* [*indepvars*] [`if` *exp*] [`in` *range*] [`, first(`*varlist*`) n1(`*vecname*`) n2(#) var(#) coding(#)` ]

`optbud` *depvar* [*indepvars*] [`if` *exp*] [`in` *range*] [`, first(`*varlist*`) prev(`*vecname*`) b(#) c1(#) c2(#)`
`var(#) coding(#)` ]

`optprec` *depvar* [ *indepvars* ] [`if` *exp*] [`in` *range*] [`, first(`*varlist*`) prev(`*vecname*`) prec(#) c1(#) c2(#)`
`var(#) coding(#)` ]

`coding` *depvar* [*first_stage_vars*]

## Description

We provide optimal sampling designs for three different scenarios. Each of these commands requires as input some pilot data, with each stratum represented by more than two observations. Such a stratified random sample is correctly handled by the mean score algorithm, called in the background, which uses the first-stage sample sizes or prevalences (also supplied by the user) to correctly weight the analysis.

The `optfixn` command calculates the optimal sampling fractions at the second stage for the situation where first-stage observations are already available and the total second-stage sample size has been decided. Such studies might arise in medical research where a database of demographic particulars on study subjects is available and expensive data (such as laboratory or radiology measurements) are to be collected for a subsample. Before running the `optfixn` command, we strongly advise running the `coding` command to see the order in which the vector of first-stage sample sizes for the various strata must be supplied. `coding` creates a variable called `grp_yz` that identifies the groups formed by the various levels of response variable $y$ and first-stage covariates $z$. In the call to `optfixn`, the first-stage sample sizes must be supplied in the same order as `grp_yz`, that is, the first element of the vector is the first-stage sample size for `grp_yz` $= 1$, the second element is for `grp_yz` $= 2$, and so on.

The `optbud` command calculates the total number of study observations and the second-stage sampling fractions that will maximize precision subject to an available budget. The user must also supply the unit cost of observations at the first and second stage. This command is applicable to the situation where a study is being planned, but the total study size has not yet been decided. Instead of first-stage sample sizes, this command expects a vector of prevalences (or estimated prevalences) for the various strata. Again, we advise running `coding` first so that these prevalences are provided in the correct order.

The `optprec` command applies to the same scenario as `optbud`, where the total sample size is not yet decided. The objective in this case is to calculate the total number of study observations and the second-stage sampling fractions that will achieve a specified precision at minimum cost. As with the `optbud` command, `optprec` expects a vector of prevalences (or estimated prevalences) for the various strata, and it is advisable to run the `coding` command first to see the order in which these values should be supplied.

Notice that each of the `optfixn`, `optbud`, and `optprec` commands have an option `coding`, which can be used if one is sure of the order in which the vector of first-stage sample sizes or prevalences should be entered. This option results in `coding` being automatically called from inside the optimal sampling command. Since this results in the creation of variables named `grp_yz` and `grp_z`, an error message will be generated if one already has variables with these names.

## Options

`first(`*varlist*`)` specifies the first-stage covariates.

`n1(`*vecname*`)` specifies the vector of first-stage sample sizes for each stratum.

`prev(`*vecname*`)` specifies the vector of prevalences for each stratum.

`n2(#)` specifies the second-stage sample sizes (used only with `optfixn`).

`b(#)` specifies the available budget (used only with `optbud`).

`c1(#)` specifies the cost per observation at the first stage (used with `optbud` and `optprec`).

`c2(#)` specifies the cost per observation at the second stage (used with `optbud` and `optprec`).

`var(#)` specifies the position in the logistic regression model of the covariate whose variance is to be minimized (that is, optimized). For example, in the simple model $Y = b_0 + b_1X_1 + b_2X_2$, if we want to minimize the variance of $X_1$, then `var` $= 2$.

`prec(#)` specifies the desired precision, that is, the variance (used only with `optprec`).

`coding(#)` is a logical flag; the default of 0 (that is, false) means that prior to calling `optfixn`, `optbud`, or `optprec` one must have run the `coding` command.

## Example 1

The following example is from CASS (Coronary Artery Surgery Study) and appears in Reilly (1996). This study collected data on the operative mortality and various risk factors for 8,096 subjects. Let us suppose that at the first stage we have only mortality status $Y$ and sex $Z$ as specified in the table below, and that it has been agreed to record the age for a subsample of 1,000 subjects in order to estimate the sex-adjusted odds ratio for age. The example is fictitious as we do have all the covariates on all subjects, but for illustrative purposes we ignore this information (that is, set values to missing). In order to compute optimal sample sizes, we require pilot data in all of the strata of the table, and so we "sampled" (reset the missing values to the actual age values) for a randomly selected 25 observations from each stratum. The resulting dataset of 100 observations is available as `pilotcas` accompanying this insert.

| | | male | female |
|---|---|---|---|
| | $Y$ | $Z = 0$ | $Z = 1$ |
| alive | $Y = 0$ | 6,666 | 1,228 |
| deceased | $Y = 1$ | 144 | 58 |

We start by computing the optimal allocation for a second-stage sample of 1,000.

```
. use pilotcas
. coding mort sex
   grp_yz      mort       sex      grp_z       nobs
        1         0         0          1         25
        2         0         1          2         25
        3         1         0          1         25
        4         1         1          2         25
for functions requiring first stage sample sizes/prevalences
enter these in the order of grp_yz
```

The coding function tells us that we have to enter the vector of first-stage sample sizes in the order specified in the following table.

| | | |
|---|---|---|
| First element | `grp_yz` $= 1$ | first-stage sample sizes for living (`mort` $= 0$) males (`sex` $= 0$) |
| Second element | `grp_yz` $= 2$ | first-stage sample sizes for living females |
| Third element | `grp_yz` $= 3$ | first-stage sample sizes for deceased (`mort` $= 1$) males |
| Fourth element | `grp_yz` $= 4$ | first-stage sample sizes for deceased females |

We enter the vector of first-stage sample sizes as follows (note the transform operator is essential).

```
. matrix fstsamp=(6666, 1228, 144, 58)´
```

Assuming the objective is to fit the logistic regression model

$$\text{logit}_i = \beta_0 + \beta_1 \text{sex}_i + \beta_2 \text{age}_i$$

and to minimize the variance of age, then we can obtain the optimal second-stage sample sizes.

```
. optfixn mort sex age, first(sex) n1(fstsamp) n2(1000) var(3)
the second stage sample sizes
----------+-----------
group(mor |
t sex)    |      Freq.
----------+-----------
        1 |          25
        2 |          25
        3 |          25
        4 |          25
----------+-----------
please check the sample sizes!

grp_yz   mort    sex  grp_z      n1      n2_pilot
     1      0      0      1    6666      25
     2      0      1      2    1228      25
     3      1      0      1     144      25
     4      1      1      2      58      25

the optimal sampling fraction(sample size) for grp_yz 1 = .089 (596)
the optimal sampling fraction(sample size) for grp_yz 2 = .164 (202)
the optimal sampling fraction(sample size) for grp_yz 3 = 1 (144)
the optimal sampling fraction(sample size) for grp_yz 4 = 1 (58)
the minimum variance for age : .00008027
Total second stage sample size =1000
```

Note that these results tell us that to minimize the variance of `age`, we need to sample all the available cases, 8.9% of controls in stratum 1 and 16.4% of controls in stratum 2.

## Example 2

This second example also uses the CASS data. Let us suppose that we wish to set up a two-stage study where at the first stage we will collect only the patient's operative mortality, sex, and weight, while at the second stage we will collect the following variables only for a subset of the study subjects.

1. Age of patients when they underwent bypass surgery.
2. The angina status of the patients when they underwent bypass surgery.
3. CHF score, that is, congestive heart failure score.
4. LVEDBP, that is, left ventricular end diastolic blood pressure.
5. Urgency of the surgery (1 for urgent, 0 for nonurgent).

Let us suppose that we have a budget of £10,000 available, that the cost of collecting data on one subject is £2 at the first stage and £15 at the second stage, and that we would like to minimize the variance of `LVEDBP` in the logistic regression model

$$\text{logit}_i = \beta_0 + \beta_1 \texttt{sex}_i + \beta_2 \texttt{weight}_i + \beta_3 \texttt{age}_i + \beta_4 \texttt{angina}_i + \beta_5 \texttt{chf}_i + \beta_6 \texttt{lvedbp}_i + \beta_7 \texttt{surgery}_i$$

As before, we need to sample a few pilot second-stage observations from each stratum defined by the different levels of mortality $Y$ and first stage covariates (`sex` and `weight`). Since first-stage covariates must be categorical, we first created a three-category weight variable `wtcat` as 1 for `weight` $< 60$, 2 for $60 \leq$ `weight` $< 70$, and 3 for `weight` $\geq 70$. The first-stage sample sizes

in the resulting 12 (2x2x3) different strata are given in the following table.

| sex | weight | alive | deceased |
|---|---|---|---|
| male | <60 | 160 | 8 |
| | 60–70 | 1,083 | 33 |
| | >70 | 5,418 | 103 |
| female | <60 | 440 | 18 |
| | 60–70 | 407 | 26 |
| | >70 | 378 | 14 |

We decided to sample 10 pilot observations from each stratum. One stratum had only 8 observations available, so all of these were included in the pilot sample. The resulting pilot data are available in the dataset `wtpilot.dta`, which can be loaded as follows and the `coding` command run to see in which order we should enter the vector of prevalences for the strata.

```
. use wtpilot
. coding mort sex wtcat
   grp_yz        mort      sex     wtcat      grp_z        nobs
        1           0          0          1          1          10
        2           0          0          2          2          10
        3           0          0          3          3          10
        4           0          1          1          4          10
        5           0          1          2          5          10
        6           0          1          3          6          10
        7           1          0          1          1           8
        8           1          0          2          2          10
        9           1          0          3          3          10
       10           1          1          1          4          10
       11           1          1          2          5          10
       12           1          1          3          6          10
for functions requiring first stage sample sizes/prevalences
enter these in the order of grp_yz
```

This tells us that our prevalence vector should be

```
. matrix prev=(0.02,.134,.670,.054,.05,.047,.001,.004,.013,.002,.003,.002)´
```

and we can find the design which will optimize (i.e., minimize) the variance of `lvedb` subject to a total budget of £10,000.

```
. optbud mort sex-surg,first(sex wtcat) prev(prev) var(7) b(10000) c1(2) c2(15)
the second stage sample sizes
----------+-----------
group(mor |
t sex     |
wtcat)    |      Freq.
----------+-----------
        1 |          10
        2 |          10
        3 |          10
        4 |          10
        5 |          10
        6 |          10
        7 |           8
        8 |          10
        9 |          10
       10 |          10
       11 |          10
       12 |          10
----------+-----------
please check the sample sizes!
grp_yz   mort    sex     wtcat  grp_z    prev      n2_pilot
     1      0      0         1      1     .02      10
     2      0      0         2      2    .134      10
     3      0      0         3      3     .67      10
     4      0      1         1      4    .054      10
     5      0      1         2      5     .05      10
     6      0      1         3      6    .047      10
```

```
     7      1      0         1      1    .001       8
     8      1      0         2      2    .004      10
     9      1      0         3      3    .013      10
    10      1      1         1      4    .002      10
    11      1      1         2      5    .003      10
    12      1      1         3      6    .002      10
the optimal sampling fraction (sample size) for grp_yz 1 = .118 (7)
the optimal sampling fraction (sample size) for grp_yz 2 = .231 (87)
the optimal sampling fraction (sample size) for grp_yz 3 = .044 (82)
the optimal sampling fraction (sample size) for grp_yz 4 = .145 (22)
the optimal sampling fraction (sample size) for grp_yz 5 = .079 (11)
the optimal sampling fraction (sample size) for grp_yz 6 = .119 (16)
the optimal sampling fraction (sample size) for grp_yz 7 = 1 (3)
the optimal sampling fraction (sample size) for grp_yz 8 = 1 (11)
the optimal sampling fraction (sample size) for grp_yz 9 = 1 (36)
the optimal sampling fraction (sample size) for grp_yz 10 = 1 (6)
the optimal sampling fraction (sample size) for grp_yz 11 = 1 (8)
the optimal sampling fraction (sample size) for grp_yz 12 = 1 (6)
the optimal number of obs = 2799
the minimum variance for lve : .00038298
total budget spent:  10023
```

Note that the optimal design samples all available cases and a varying proportion of controls in the different sex−weight categories.

### Example 3

In Example 2, we used the `optbud` command to find an optimal design subject to a budget of £10,000, where the cost per first-stage observation was £2 and the cost per second-stage observation was £15. The minimum achievable variance for the variable `lvedbp` was .00038298.

Now we reverse our question. If we wish to achieve a variance of .00038298 for `lvedbp`, what is the design that will minimize the study cost? The function `optprec` calculates the design to minimize the cost subject to a desired precision, and so can be used to answer this question.

```
. use wtpilot
. coding mort sex wtcat
  (output omitted)
. matrix prev=(0.02,.134,.670,.054,.05,.047,.001,.004,.013,.002,.003,.002)´
. optprec mort sex-surg,first(sex wtcat) prev(prev) var(7) prec(.00038298) c1(2) c2(15)
  (output omitted)
```

The optimal design for this case is exactly the same as its counterpart in Example 2, as these are simply two ways of asking the same question.

### References

Reilly, M. 1996. Optimal sampling strategies for two-stage studies. *American Journal of Epidemiology* 143: 92–100.

Reilly, M. and M. S. Pepe. 1995. A mean score method for missing and auxiliary covariate data in regression models. *Biometrika* 82: 299–314.

## sxd3 | Sample size for the kappa-statistic of interrater agreement

Michael E. Reichenheim, Instituto de Medicina Social/UERJ, Brazil, michael@ims.uerj.br

**Abstract:** The dialog-box-driven program `sskdlg` for calculating the sample size for the kappa-statistic when there are two unique raters evaluating a binary event is introduced and illustrated.

**Keywords:** sample size, kappa statistics, dialog box.

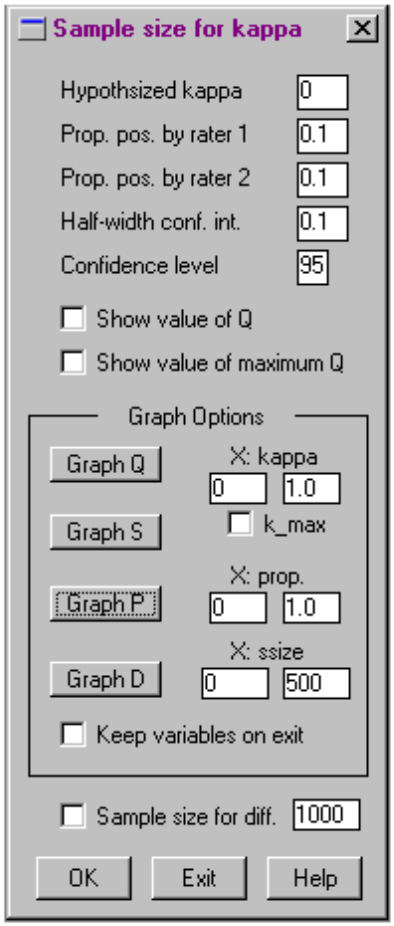
### Introduction

In recent years there has been an increasing call for researchers in the fields of psychiatry and epidemiology to account for the stochasticity of reliability estimators, among them the kappa-statistic measure of interrater agreement (Shrout and Newman 1989). Yet, if this issue is to be addressed properly, calculating sample sizes in the planning stage of an investigation becomes mandatory. Although some proposals for calculating sample size are available in the literature (Linnet 1987, Donner and Eliasziw 1992, Cantor 1996, Walter et al. 1998; Shrout and Newman 1989), to our knowledge there has only been a limited implementation in one sample size oriented package (Statistical Solutions 1999) and none in any of the major commercial statistical software packages, Stata included.

This article presents a dialog-box-driven program `sskdlg` to calculate the sample size for the kappa statistic when there are two unique raters evaluating a binary event. `sskdlg` is geared towards calculating a sample size from a precision oriented perspective, that is, choosing a sample size so that the standard error of the estimate and the resulting limits for a confidence interval do not exceed specified values. The program is based on the asymptotic variance presented by Fleiss, et al. (1969) (see also Fleiss 1981, equations 13.15–13.18) and follows the procedure outlined by Cantor (1996). This procedure is based on a quantity

$$Q = (1-\pi_e)^{-1}\Big\{\sum_i \pi_0\left[(1-\pi_e) - (\pi_{.i}+\pi_{i.})(1-\pi_0)\right]^2 + (1-\pi_0)^2\sum_{i\neq j}\pi_{ij}(\pi_{.j}+\pi_{j.}^2) - (\pi_0\pi_e - 2\pi_e + \pi_0)^2\Big\}$$

where, given a $2\times 2$ table, $\pi_e = \pi_{1.}\pi_{.1} + \pi_{2.}\pi_{.2}$ and $\pi_o = \pi_{11} + \pi_{22}$. Since $Q$ equals the variance of kappa times the sample size, the latter can be solved out and calculated.




The dialog box is presented above. Without any options, clicking the "OK" button displays the calculated sample size in the Results window. The following five parameters are needed.

$\kappa$ The estimate of kappa the researcher expects to find. `sskdlg` uses the default value of 0 which presumes calculating a standard error when both ratings are independent. Although more realistic values are possible and should be encouraged, this default value, albeit extremely conservative, is suitable for projecting the sample size of a study that, ultimately, would be analyzed using the traditional equation for the standard error as found in Stata's `kap` program. Depending on the specification of other conditions, the expected kappa can take any value within the permissible bounds, never exceeding $-1$ to 1. If the specified kappa is incompatible with the selected marginals (proportion of positives expected by each rater) or is outside the plausible range, `sskdlg` outputs a warning and the sample size is not calculated.

$p_1$ The proportion of study subjects one expects the first observer to rate as positives (or the 1st observation in the case or a test-retest reproducibility study). `sskdlg` uses the default value of 0.1 when the dialog box is opened.

$p_2$ The proportion of study subjects one expects the second observer to rate as positives (or observation). `sskdlg` uses the same default value as for $p_1$.

$d$ The envisaged absolute precision, i.e., the difference between kappa and its lower (or upper) $1-\alpha$ confidence limit. Acceptable values range from $> 0$ to a limit where, given a set of marginal values ($p_1$ and $p_2$), values of $d$ entailing $\kappa - d$ below the minimum possible value of kappa ($\kappa_{\text{min}}$) or $\kappa + d$ above the maximum possible value of kappa ($\kappa_{\text{max}}$) are disallowed. If both conditions are satisfied, `sskdlg` returns a message warning that the sample size will not be calculated. If only one of those occurs, the user is warned that there is a partial incompatibility between the stated values of $d$ and $\kappa$. Yet, the sample size is calculated since $d$ is compatible with at least one of $\kappa$'s boundary values.

CI Confidence level percent for the confidence interval. The default is 95%.

Several display and graphical options are available. Checking "Show value of Q" and "Show value of maximum Q" adds those values to the sample size displayed in the Results window. The first check is simply the quantity that underlies the sample size calculation. The second check requests the calculation of the largest possible value of $Q$ and the corresponding sample size. This is important when the researcher is not prepared to make any prior assumption concerning kappa. The output also indicates the maximum possible value of kappa, given the preset marginals $p_1$ and $p_2$.

Checking "Sample size for diff." and filling in the desired value requests a unique value of the absolute precision ($d$), given the other selected inputs. This enables the user to work backwards by finding out the precision corresponding to a preset sample size.

There are 4 types of graphics that can be selected. Clicking the "Graph Q" button requests a graphical display of the $Q$ values according to a range of kappa values. The default range is 0 to 1.0 when the dialog box is opened. The editing boxes (e.g., indicated by "X:kappa") can be used to specify a desired range of values for the $X$-axis. This operates as a zooming device. Changing values enables zooming in or out. Note that negative values are allowed although this should be unusual in the context of reliability studies. The position of maximum possible value of kappa ($\kappa_{\text{max}}$) can also be visualized in the graph by checking "k_max". Also note that when the specified parameters preclude the calculation of sample size, Graph Q will not be (re)displayed. Values need to be reset in order to enable the graph.

Clicking the "Graph S" button requests a graphical display analogous to Graph Q but plots sample size instead. The same editing boxes as for Graph Q to control the $x$-axis (zooming) are used. "k_max" may also be checked. The same restrictions as in Graph Q apply here too.

Clicking the "Graph P" button requests a graphical display of sample sizes according to the proportion of positives measured by raters 1 and 2 when both are expected to find the same value and given the specified values of $\kappa$, $d$ and CI. The $x$-axis default range is 0 to 1.0. Nevertheless, Graph P will only show a range compatible with plausible sample sizes, since some combinations of specified parameters are impossible. For zooming in or out, edit boxes indicated by "X:prop." can be used.

Clicking the "Graph D" button requests a graphical display of the absolute precision for a range of prespecified sample sizes. This display is an extension of "Sample size for diff.". Sample size range (zooming) can be controlled using the edit boxes indicated by "X:ssize".

Finally, on leaving the dialog box, checking "Keep variables on exit" retains in memory essential variables used for displaying Graphs Q, S, P and D. This enables the user to redraw new graphs at his/her own discretion. Note that values kept in memory are those specified (on display) at the time of exit. This option requires at least running the program once or running a viable configuration of parameters after an improper one precluded a calculation. This is because the underlying data is cleared in this situation in order to avoid a mismatch between the parameters on screen (dialog box) and the data in memory generated by a former viable run.

(*Continued on next page*)

## Examples

Typing `sskdlg` in the Command window calls the dialog box. Without checking the options or changing the default values, the following output is displayed on pressing the OK button:

```
--- Begin ------------------------------------------------
  Results for kappa=0, p1=0.1, p2=0.1,
  d=0.1 [95% Conf. Interval]:

   *  Sample size = 384
--- End --------------------------------------------------
```

Changing the input parameters and checking all options will produce the following display

```
--- Begin ------------------------------------------------
  Results for kappa=0.1, p1=0.2, p2=0.2,
  d=0.2 [95% Conf. Interval]:
    * Sample size = 114
    *  Value of Q = 1.182

  Given the values specified above, the
  maximum sample size is 126 for a
  kappa of .289

  If the sample size is fixed at 50,
  given the kappa, p1, p2 and CI stated
  above, d = .215.
--- End --------------------------------------------------
```

The above set of statements are worth commenting on from a practical view point. The first indicates that if the researcher states that a) both raters are expected to find a prevalence of 20% of the event of interest, b) the null-hypothesis for $\kappa$ is 0.1, and c) he or she is ready to tolerate a rather lenient absolute precision of 0.2, given a 95% confidence interval, this reliability study will need at least a sample size of 114 subjects.

The second statement conveys that if the researcher is not prepared to make any assumption about the value of $\kappa$ , i.e., to assert a null hypothesis for the parameter, the most stringent situation (given all the other parameters he or she has specified) would occur when $\kappa = 0.289$ for which a sample size of 126 would be needed.

The last output says that, given the other specified values, if resources or logistics allow for just 50 subjects in a reliability study, the absolute precision would be 0.215. This, in turn, is only slightly worse than the value specified in the first place (0.2), which means that the researcher may decide to go ahead with his or her "half sized" study since not much precision would have been lost.

As has been hinted so far, there are many situations that arise from irregular or incompatible specifications. In those circumstances, `sskdlg` precludes any unwarranted output. For instance, the output below follows a possible mishap (value of $d$ relative to $\kappa$), displaying in red the message

```
The specified value of d (0.8) is incompatible
with the kappa you selected (0.5), since
both confidence limits (-.3 and 1.3)
exceed the possible boundary values for kappa
(-.11 and 1, respectively), given the
specified marginals (p1 = 0.1 and p2 = 0.1).
Sample size will not be calculated.
```

Turning to some graphical outputs, Figure 2 shows a "Graph S" for the parameters specified above. Note that this graph has been constrained to kappa values between 0 and 0.7. Sample size starts at 96, peaks at 126 where $\kappa_{\max} = 0.289$, and decreases to 78 at $\kappa = 0.7$. Also note that "k_max" has been checked and a vertical line positioned at $\kappa_{\max}$.

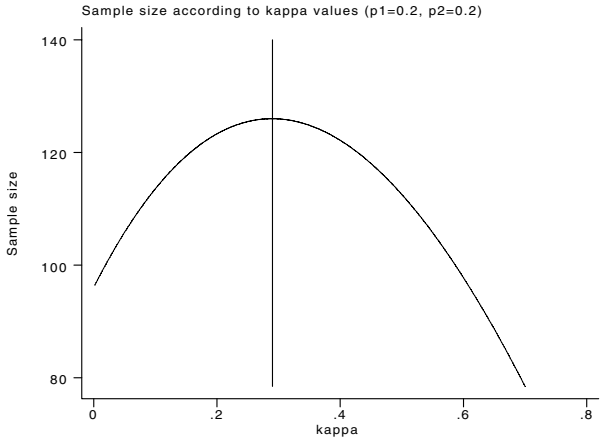



Figure 2. An example of using "Graph S".

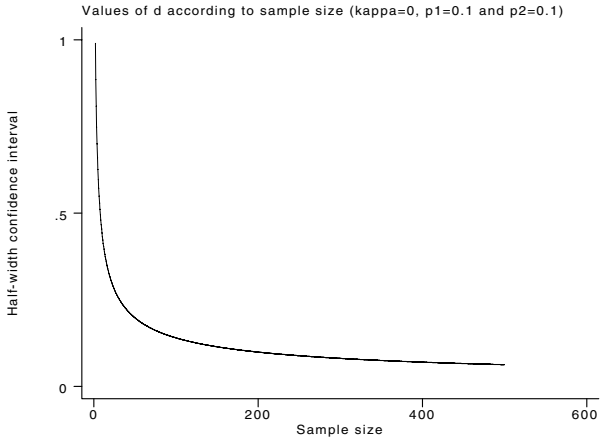



Figure 3. An example using "Graph D".

Figure 3 illustrates a "Graph D" output for the default values.

Note that the graph's default values of 0 to 500 were left unchanged in the "X:ssize" edit boxes. Yet, for the parameters at hand, this graph is not very informative since the sample size spectrum where the decline of most precision values takes place is quite small. The researcher would need a much narrower sample size range over which the $d$ values decrease less steeply. This enhanced picture would allow him/her to make a better decision, reaching a compromise between a viable sample size and an acceptable precision for the kappa estimates. This zooming is illustrated in Figure 4 where the "X:ssize" edit boxes have been changed to 50 to 250. Note the tighter range of the $d$ values on the vertical axis, now comprising a more realistic set of figures to assess precision.

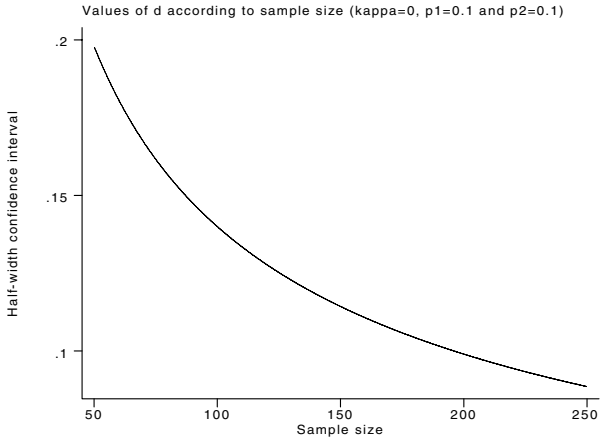



Figure 4. Using the zoom feature.

## References

Cantor, A. B. 1996. Sample size calculations for Cohen's k. *Psychological Methods* 1: 150–153.

Donner, A. and M. Eliasziw. 1992. A goodness-of-fit approach to inference procedures for the kappa statistic: confidence interval construction, significance-testing and sample size estimation. *Statistics in Medicine* 11: 1511–1519.

Fleiss, J. L., 1981. *Statistical Methods for Rates and Proportions.* 2d ed. New York: John Wiley & Sons.

Fleiss, J. L., J. Cohen, and B. S. Everitt. 1969. Large sample standard errors for kappa and weighted kappa. *Psychological Bulletin* 72: 323–327.

Linnet, K. 1987. Comparison of quantitative diagnostic tests: type I error, power, and sample size. *Statistics in Medicine* 6: 147–158.

Shrout, P. E. and S. C. Newman. 1989. Design of two-phase prevalence surveys of rare disorders. *Biometrics* 45: 549–555.

*Statistical Solutions.* 1999. nQuery Advisor 3.0. Saugus, MA.

Walter, S. D., M. Eliasziw, and A. Donner. 1998. Sample size and optimal design for reliability studies. *Statistics in Medicine* 17: 101–110.

## STB categories and insert codes

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* | datasets | *qs* | questions and suggestions |
| *gr* | graphics | *tt* | teaching |
| *in* | instruction | *zz* | not elsewhere classified |

*Statistical Categories:*

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

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

## Guidelines for authors

The Stata Technical Bulletin (STB) is a journal that is intended to provide a forum for Stata users of all disciplines and levels of sophistication. The STB contains articles written by StataCorp, Stata users, and others.

Articles include new Stata commands (ado-files), programming tutorials, illustrations of data analysis techniques, discussions on teaching statistics, debates on appropriate statistical techniques, reports on other programs, and interesting datasets, announcements, questions, and suggestions.

A submission to the STB consists of

1. An insert (article) describing the purpose of the submission. The STB is produced using plain TEX so submissions using TEX (or LATEX) are the easiest for the editor to handle, but any word processor is appropriate. If you are not using TEX and your insert contains a significant amount of mathematics, please FAX (979–845–3144) a copy of the insert so we can see the intended appearance of the text.

2. Any ado-files, `.exe` files, or other software that accompanies the submission.

3. A help file for each ado-file included in the submission. See any recent STB diskette for the structure a help file. If you have questions, fill in as much of the information as possible and we will take care of the details.

4. A do-file that replicates the examples in your text. Also include the datasets used in the example. This allows us to verify that the software works as described and allows users to replicate the examples as a way of learning how to use the software.

5. Files containing the graphs to be included in the insert. If you have used STAGE to edit the graphs in your submission, be sure to include the `.gph` files. Do not add titles (e.g., "Figure 1: ...") to your graphs as we will have to strip them off.

The easiest way to submit an insert to the STB is to first create a single "archive file" (either a `.zip` file or a compressed `.tar` file) containing all of the files associated with the submission, and then email it to the editor at `stb@stata.com` either by first using `uuencode` if you are working on a Unix platform or by attaching it to an email message if your mailer allows the sending of attachments. In Unix, for example, to email the current directory and all of its subdirectories:

```
tar -cf - . | compress | uuencode xyzz.tar.Z > whatever
mail stb@stata.com < whatever
```

## International Stata Distributors

International Stata users may also order subscriptions to the *Stata Technical Bulletin* from our International Stata Distributors.

| | | | |
|---|---|---|---|
| Company: | Applied Statistics & Systems Consultants | Countries served: | Israel |
| | | Phone: | +972 (0)6 6100101 |
| Address: | P.O. Box 1169 | Fax: | +972 (0)6 6554254 |
| | 17100 NAZERATH-ELLIT, Israel | Email: | assc@netvision.net.il |
| Company: | Axon Technology Company Ltd | Countries served: | Taiwan |
| Address: | 9F, No. 259, Sec. 2 | Phone: | +886-(0)2-27045535 |
| | Ho-Ping East Road | Fax: | +886-(0)2-27541785 |
| | TAIPEI 106, Taiwan | Email: | hank@axon.axon.com.tw |
| Company: | Chips Electronics | Countries served: | Indonesia, Brunei, Malaysia, Singapore |
| Address: | Kelapa Puyuh IV KB 23 | Phone / Fax: | 62 - 21 - 452 17 61 |
| | Kelapa Gading Permai | Mobile Phone: | 62 - 81 - 884 95 84 |
| | JAKARTA 14240, Indonesia | Email: | puyuh23@indo.net.id |
| Company: | Dittrich & Partner Consulting | Countries served: | Germany, Austria, Czech Republic, Hungary, Poland |
| Address: | Kieler Straße 17 | | |
| | 5. floor | Phone: | +49 2 12 / 26 066 - 0 |
| | D-42697 Solingen, Germany | Fax: | +49 2 12 / 26 066 - 66 |
| URL: | http://www.dpc.de | Email: | sales@dpc.de |
| Company: | IEM | Countries served: | South Africa, Botswana, Lesotho, Namibia, Mozambique, Swaziland, Zimbabwe |
| Address: | P.O. Box 2222 | | |
| | PRIMROSE 1416, South Africa | | |
| | | Phone: | +27-11-8286169 |
| | | Fax: | +27-11-8221377 |
| | | Email: | iem@hot.co.za |
| Company: | JasonTech Inc. | Countries served: | Korea |
| Address: | 181-3 Hansang B/D, Bangyidong | Phone: | +82-(0)2-420-6700 |
| | Songpaku | Fax: | +82-(0)2-420-8600 |
| | Seoul 138-050, Korea | Email: | info@jat.co.kr |
| Company: | MercoStat Consultores | Countries served: | Uruguay, Argentina, Brazil, Paraguay |
| Address: | 9 de junio 1389 | Phone: | 598-2-613-7905 |
| | CP 11400 MONTEVIDEO, Uruguay | Fax: | 598-2-613-7905 |
| | | Email: | mercost@adinet.com.uy |
| Company: | Metrika Consulting | Countries served: | Sweden, Baltic States, Denmark, Finland, Iceland, Norway |
| Address: | Mosstorpsvagen 48 | | |
| | 183 30 Taby STOCKHOLM, Sweden | Phone: | +46-708-163128 |
| URL: | http://www.metrika.se | Fax: | +46-8-7924747 |
| | | Email: | sales@metrika.se |
| Company: | MultiON Consulting, SA de CV | Countries served: | Mexico |
| Address: | Insurgentes Sur 1236-301 | Phone: | 52 (5) 559-4050 Ext 190 |
| | Mexico, DF, 03200, Mexico | Fax: | 52 (5) 559-4048 |
| | | Email: | multion@multion.com.mx |

(*List continued on next page*)

## International Stata Distributors

(*Continued from previous page*)

| | | | |
|---|---|---|---|
| Company: | Ritme Informatique | Countries served: | France, Belgium, Luxembourg |
| Address: | 34, boulevard Haussmann | Phone: | +33 (0)1 42 46 00 42 |
| | 75009 Paris, France | Fax: | +33 (0)1 42 46 00 33 |
| URL: | http://www.ritme.com | Email: | info@ritme.com |
| Company: | Scientific Solutions S.A. | Countries served: | Switzerland |
| Address: | Avenue du Général Guisan, 5 | Phone: | 41 (0)21 711 15 20 |
| | CH-1009 Pully/Lausanne, Switzerland | Fax: | 41 (0)21 711 15 21 |
| | | Email: | info@scientific-solutions.ch |
| Company: | Smit Consult | Countries served: | Netherlands |
| Address: | Doormanstraat 19 | Phone: | +31 416-378 125 |
| | 5151 GM Drunen, Netherlands | Fax: | +31 416-378 385 |
| URL: | http://www.smitconsult.nl | Email: | info@smitconsult.nl |
| Company: | Survey Design & Analysis Services Pty Ltd | Countries served: | Australia, New Zealand |
| Address: | PO Box 1206 | Phone: | +61 (0)3 9878 7373 |
| | Blackburn North VIC 3130, Australia | Fax: | +61 (0)3 9878 2345 |
| URL: | http://survey-design.com.au | Email: | sales@survey-design.com.au |
| Company: | Timberlake Consultants | Countries served: | United Kingdom, Eire |
| Address: | Unit B3 Broomsleigh Bus. Park | Phone: | +44 (0)208 697 3377 |
| | Worsley Bridge Road | Fax: | +44 (0)208 697 3388 |
| | LONDON SE26 5BN, United Kingdom | Email: | info@timberlake.co.uk |
| URL: | http://www.timberlake.co.uk | | |
| Company: | Timberlake Consultants Srl | Countries served: | Italy |
| Address: | Via Baden Powell, 8 | Phone: | +39 (0)864 210101 |
| | 67039 SULMONA (AQ), Italy | Fax: | +39 (0)864 206014 |
| URL: | http://www.timberlake.it | Email: | timberlake@arc.it |
| Company: | Timberlake Consulting S.L. | Countries served: | Spain |
| Address: | Calle Mendez Nunez, 1, 3 | Phone: | +34 (9) 5 422 0648 |
| | 41011 Sevilla, Spain | Fax: | +34 (9) 5 422 0648 |
| | | Email: | timberlake@zoom.es |
| Company: | Timberlake Consultores, Lda. | Countries served: | Portugal |
| Address: | Praceta Raúl Brandao, n°1, 1°E | Phone: | 351 (0)1 471 73 47 |
| | 2720 ALFRAGIDE, Portugal | Fax: | +351 (0)1 471 73 47 |
| | | Email: | timberlake.co@mail.telepac.pt |
| Company: | Vishvas Marketing-Mix Services | Countries served: | India |
| Address: | C\O S. D. Wamorkar | Phone: | +91-251-440087 |
| | "Prashant" Vishnu Nagar, Naupada | Fax: | +91-22-5378552 |
| | THANE - 400602, India | Email: | vishvas@vsnl.com |