View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Research Papers in Economics

STATA TECHNICAL BULLETIN

January 1999

STB-47

A publication to promote communication among Stata users

Editor

H. Joseph Newton
Department of Statistics
Texas A & M University
College Station, Texas 77843
409-845-3142
409-845-3144 FAX
stb@stata.com EMAIL

Associate Editors

Nicholas J. Cox, University of Durham
Francis X. Diebold, University of Pennsylvania
Joanne M. Garrett, University of North Carolina
Marcello Pagano, Harvard School of Public Health
J. Patrick Royston, Imperial College School of Medicine

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

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

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

Copyright Statement. The Stata Technical Bulletin (STB) and the contents of the supporting files (programs, datasets, and help files) are copyright © by StataCorp. The contents of the supporting files (programs, datasets, and help files), may be copied or reproduced by any means whatsoever, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the STB.

The insertions appearing in the STB may be copied or reproduced as printed copies, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the STB. Written permission must be obtained from Stata Corporation if you wish to make electronic copies of the insertions.

Users of any of the software, ideas, data, or other materials published in the STB or the supporting files understand that such use is made without warranty of any kind, either by the STB, the author, or Stata Corporation. In particular, there is no warranty of fitness of purpose or merchantability, nor for special, incidental, or consequential damages such as loss of profits. The purpose of the STB is to promote free 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

an67	Stata 5, Stata 6, and the STB

Stata 6 has been released and this is the last issue of the STB that is explicitly Stata 5, which is to say, every insert in this issue will work equally well with Stata 5 or Stata 6. Future issues will contain inserts based on Stata 5 code and Stata 6 code and, over time, one would expect nearly all inserts to be in terms of Stata 6.

Those of you using Stata 6 for Windows 98/95/NT, Stata for PowerMac, and Stata for Unix, and who have access to the Internet, can now obtain the programs corresponding to the STB inserts published here from inside Stata: Either

pull down **Help**, select **STB and User-written Programs**, click on *http://www.stata.com*, then click on *stb*;

or

type `net from http://www.stata.com`, and then type `net cd stb`.

All issues are available, from STB–1 to this one.

You can also obtain the official updates to Stata over the net:

pull down **Help**, select **Official Updates**, and then click on *http://www.stata.com*;

or

type `update from http://www.stata.com`.

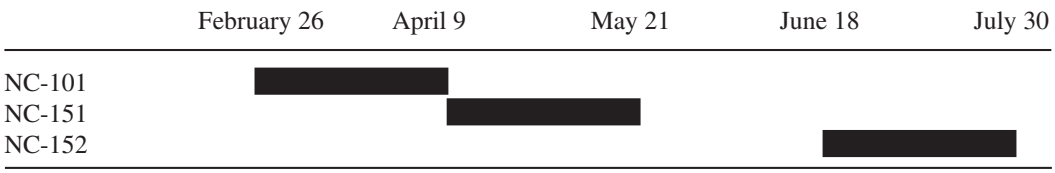
For those of you without web access and who receive the STB with diskettes, both Stata 5 format and Stata 6 format diskettes are included with this issue of the STB. Next issue, only the Stata 6 format diskettes will be included.

On the Stata 5 format diskettes, you will find that there are no official updates. The last changes to Stata 5 were made last issue.

On the Stata 6 format diskettes, in addition to programs corresponding to the STB inserts published here, there are updates for Stata 6. Installing official updates and STB inserts is easy: Windows users should see Chapter 20 in their *Getting Started* manual, Macintosh users should see Chapter 20 in their *Getting Started with Stata for Macintosh* manual, and Unix users should see Chapter 17 in their *Getting Started with Stata for Unix* manual.

an68	NetCourse schedule announced

We have announced our latest NetCourse schedule:



NetCourse 101. An introduction to Stata

6 weeks (4 lectures)	
Course dates:	February 26 through April 9
Deadline for enrollment:	February 22
Cost:	$ 85
Course leaders:	Mark Inlow and Mark Esman
Prerequisites:	Stata 6, installed and working
Schedule:	
Lecture 1	February 26
Lecture 2	March 5
One-week break	March 11 through March 17
Lecture 3	March 19
Lecture 4	March 26
Closing discussion	
Course ends	April 9

NetCourse 151. Introduction to Stata programming

6 weeks (4 lectures)	
Course dates:	April 9 through May 21
Deadline for enrollment:	April 5
Cost:	$100
Course leaders:	Ken Higbee and Mark Esman
Prerequisites:	Stata 6, installed and working
	Basic knowledge of using Stata interactively
Schedule:	
Lecture 1	April 9
Lecture 2	April 16
One-week break	April 22 through April 28
Lecture 3	April 30
Lecture 4	May 7
Closing discussion	
Course ends	May 21

NetCourse 152. Advanced Stata programming

6 weeks (4 lectures)	
Course dates:	June 18 through July 30
Deadline for enrollment:	June 14
Cost:	$100
Course leaders:	William Gould and Vince Wiggins
Prerequisites:	Stata 6, installed and working
	NetCourse 151 or equivalent knowledge
Schedule:	
Lecture 1	June 18
Lecture 2	June 25
One-week break	July 1 through July 7
Lecture 3	July 9
Lecture 4	July 16
Closing discussion	
Course ends	July 30

More information, including an outline of each course, can be obtained by

1. Pointing your browser to *http://www.stata.com*.
2. Clicking on the Headline *NetCourse schedule announced*.

Email *stata@stata.com* for enrollment forms.

NetCourses are courses offered over the Internet via email then run about 6 weeks. Every Friday a "lecture" is emailed to the course participants. After reading the lecture, participants email questions and comments back to the Course Leaders. Thses emailed questions are remailed to all course participants by the NetCourse software. Course leaders respond to the questions and comments on Tuesday and Thursday. The other participants are encouraged to amplify or otherwise respond to the questions and comments as well. The next lecture is then emailed on Friday and process repeats.

The courses are designed to take roughly 3 hours per week.

All courses have been updated to Stata 6, which amounts to minor revisions for NC-101 and NC-151, and a major revision for NC-152.

gr34 | Drawing Venn diagrams

Jens M. Lauritsen, County of Fyn, Denmark, jml@dadlnet.dk

`venndiag` produces a so-called Venn diagram based on variables in a dataset. It consists of a number of rectangles, each corresponding to one of the variables in *varlist*. The rectangles are arranged so that they overlap and delimit areas. In each area, the counts of records are shown for the relevant combination of *varlist*. With two variables `A` and `B`, the counts of records where

both variables equal one is placed in the overlapping area of A and B. In the part of A not in B, the count of variables where A is one and B is not one is placed and so on.

The command has three types of output:

- one which creates the combinations of the variables and presents the counts of this in the log file,
- one which contains the actual diagram presented on the left side of a graph window,
- and one which presents labeling and contents information on the right side of a graph window.

venndiag could be used, for example, to 1) show the number of persons having different symptoms indicated in three variables, e.g., asthma, hayfever, and eczema, 2) show in a household survey the numbers having cats, dogs, and birds, 3) count specific diagnoses placed in one or several variables, 4) combine variables and achieve frequency counts in a log file and in a new variable, and 5) create a new combined variable of 2–4 variables (with or without missing values).

Syntax

venndiag *varlist* [if *exp*] [in *range*] [, label(*str*) show(*str*) missing gen(*varnames*)
list(*variables*) print saving(*filename*) c1(#) c2(#) c3(#) c4(#) noframe
nograph nolabel t1title(*str*) t2title(*str*) t3title(*str*) r1itle(*str*)
r2title(*str*) r3title(*str*) r4itle(*str*) r5title(*str*) r6title(*str*)]

where the *varlist* must contain from 2–4 numerical variables and if generating a variable, that variable must not exist.

Options

label(*str*) allows one to specify one or more of the following items by including in *str* one or more of the following (the default is adft and order is not important):

c	variable names and value counted in each variable.
f	filename and date.
d	overall description.
m	indicates counts of missing in each variable.
t	titles are shown.
a	date of creation of graph added.
i	total records in file, # excluded (in/if or missing values) and number of records in graph shown.
x	show information in titles r1title, . . . , r6title (should not be used with fdmc).
all	short form for all labels (not x).

show(*str*) specifies one or more items to be displayed on the graph by including in *str* one or more of the following (the default is pcts and the order of items does not matter):

p	percent of area.
c	count in each area.
t	percent of each variable.
v	use variable names instead of A B C D.
f	add footnote below explaining percentages.
n	display counts and percentages for areas with counts of 0.
x	exclude the counts and or percentages of the non-area (records in non-area are still included in N).
all	short form for all (not x).

missing includes all records regardless of missing variables in *varlist*.

gen(*varnames*) adds variables named by *varnames* to the dataset. They must be nonexisting variables.

list(*variables*) lists *varlist* in call of venndiag and *variables* after combining records.

print prints the graph immediately (Windows 95 and Macintosh only).

saving(*filename*) saves the graph to *filename* (replacing file).

`c1(#)`, `c2(#)`, `c3(#)`, and `c4(#)` specifies which value to regard as an affirmative outcome in `v1`, . . . , `v4`. The default is 1. It must be an integer.

`noframe` removes the outer frame from the graph.

`nograph` means do not show graph, regardless of contents of `show()` and `label()`.

`nolabel` disregards all labeling, regardless of contents of `label`.

`t1title(`*str*`)`, `t2title(`*str*`)`, and `t3title(`*str*`)` define titles shown on the graph. Defaults are `t1title(Venn Diagram)` and `t3title(N = #records)`.

`r1title(`*str*`)`, `r2title(`*str*`)`, `r3title(`*str*`)`, `r4title(`*str*`)`, `r5title(`*str*`)`, and `r6title(`*str*`)` are additional titles to be shown when `x` is included in `label()`.

Remarks

Text in the graph is sized accordingly to the `set textsize` command, with 100 being the default. Experimentation with sizes is recommended. If set at a value above 120, some text might be outside the rectangles in the graph.

Disallowed variable names in a dataset are `_merge`, `vdx1199`, and `vd_id199`. If a system macro is set with the command (`global S_grid = "y"`) a grid will be placed on the screen for further placement of text by a user.

Information is retrievable in `S_*` variables after execution (see the start of `venndiag.ado` for the numbers of these variables).

The basis for percentages is the number of records included in the graph totally. If the user wishes percentages to be based on only those affirmative records of at least one of the variables, then include only affirmative records of at least one of the variables. Sometimes the user might want to show a graph with percentages based on all cases, but not showing the records which do not have at least one affirmative variable. This is accomplished with the `show(x)` option.

Examples

The simplest plot with no options specified will appear as in Figure 1. The boxes are named A, B, C (with three variables) and counts of each area placed in an appropriate place. Percentages of areas are shown in parentheses and for each variable the percentage having the counted outcome is shown without parentheses. Titles in the default mode with no options are as seen, the date of creation and datefile used plus variable labels and total N shown with the general title "Venn Diagram". Any record having a value of missing will be excluded from the graph. This will be indicated in the log file together with the counts of the possible combinations of the variables. To produce Figure 1, we generated four variables named `eczema`, `asthma`, `season`, and `atopi` representing the presence (represented by a 1) or absence (represented by a 0) of four diseases on each of 4,000 cases. The values of all four variables are 0 for all 4,000 cases except

Variable	Nonzero values
`eczema`	`. in 1400/1425` `1 in 1065/1075, 1326/1399, 1425/1599, 1800/1999`
`asthma`	`. in 1100/1125` `1 in 100/199, 1060/1099, 1126/1399`
`season`	`. in 1200/1225` `1 in 200/299, 1085/1199, 1250/1499`
`atopi`	`. in 1390/1415` `1 in 300/399, 1200/1389, 1416/1499.  1790/1825`

Here is a simple example using the first three variables:

(*Example on next page*)

```
. venndiag asthma season eczema
------------------------------------------------------------------------
Venn diagram of variables: asthma season eczema
File: testdata.dta (Cr:16 Nov 1998 )

     Outcome    Variable and label
A:       1           asthma    Asthma previous year
B:       1           season    Seasonal allergic symptoms
C:       1           eczema    Current hand eczema

    4000   Records in file
      78   Records excluded by missing values
    ----
    3922   Records in Diagram:
Counts for combined variables:
------------------------------------------------------------------------
A       |    138    4 %  (asthma == 1)&(season != 1)&(eczema != 1)
B       |    100    3 %  (asthma != 1)&(season == 1)&(eczema != 1)
C       |    300    8 %  (asthma != 1)&(season != 1)&(eczema == 1)
AB      |    165    4 %  (asthma == 1)&(season == 1)&(eczema != 1)
AC      |     11    0 %  (asthma == 1)&(season != 1)&(eczema == 1)
BC      |     74    2 %  (asthma != 1)&(season == 1)&(eczema == 1)
ABC     |     74    2 %  (asthma == 1)&(season == 1)&(eczema == 1)
---     |   3060   78 %  (asthma != 1)&(season != 1)&(eczema != 1)
------------------------------------------------------------------------
------------------------------------------------------------------------
```

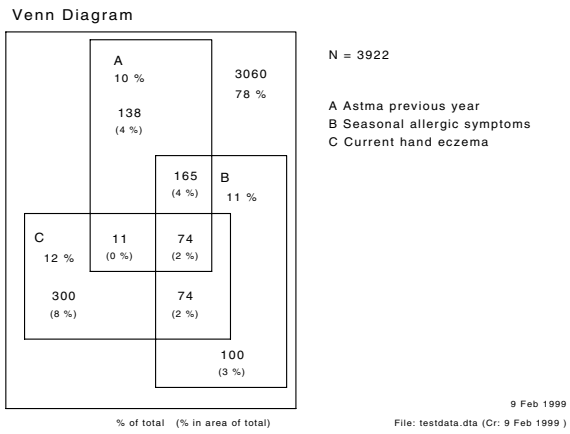


Figure 1. `venndiag` for three variables with no options.

Sometimes a variable is recorded with positive answers only, i.e., a missing indicates a "non-affirmative" answer. When this is the case, records containing missing values should be included. This is indicated by the option `missing`. In the second example a few other options are shown. Values indicated in `c1()` to `c4()` will be used to count outcomes instead of a 1, and additional information on the graph will be shown with the `label(all)` and `show(all)` options. The `all` is a short form for several options, see above. With the `noframe` option, the external frame around the boxes is not shown. The log file will inform the user of the inclusion of missing values.

```
. venndiag eczema-atopi, noframe missing  t3(Note: 0´s shown on graph)
> label(all) show(all)  t1(Venn Diagram - all information shown on right half of graph)
> c1(2) c2(3) c3(4) c4(5)
------------------------------------------------------------------------
Venn diagram of variables: eczema asthma season atopi
File: testdata.dta (Cr:16 Nov 1998 )
  (output omitted )
4000   Records in Diagram:

     104   variables in all records contain missing values
           eczema       :     26
           asthma       :     26
           season       :     26
           atopi        :     26

Counts for combined variables:
------------------------------------------------------------------------
A       |    274    7 %  (eczema == 2)&(asthma != 3)&(season != 4)&(atopi != 5)
  (output omitted )
```

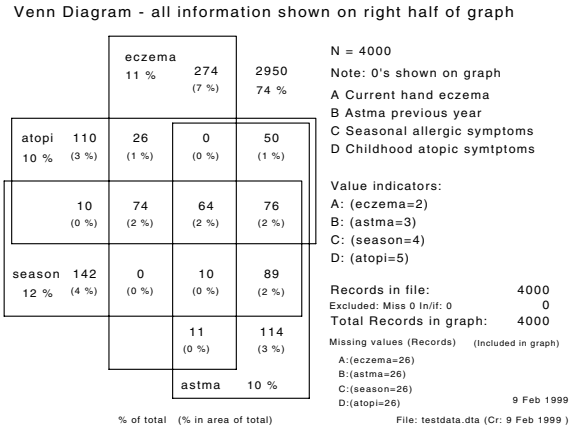


Figure 2. `venndiag` for four variables using several options.

Defaults for labels are convenient, but sometimes a fine-tuning is desirable. If the user specifies `label(xt)`, only texts put into `t1()`, ..., `t3()` and `r1()`, ..., `r6()` are added to the text. With the options in `show()` the graph contents on the left can be controlled. The bottom line description is excluded by omitting `f` from the parentheses, e.g., `show(xcpt)`. By specifying `gen(vd1)` a new variable is added to the dataset. When creating a new variable, a note is added to the dataset indicating the date and which variables with which values were applied. Additionally, any `if/in` `missing` options will be added to the note. One other possibility is to enlarge the textsize by applying the general command `set textsize`. In Figure 3, it is set to 120. The log file will indicate the creation of the new variable:

```
. set textsize 120
. venndiag asthma season in 1/3000, show(xcpt) label(tx)
>  r1(Allergi related symptoms) r2(A:eczema)  gen(vd1)
>  r3(B:asthma) r4(N=2948) r5(Text: set textsize 120) r6(t1..t3 available for other texts)
------------------------------------------------------------------------
Venn diagram of variables: asthma season
  (output omitted)
New variable created. Name: vd1  Label: asthma season(vd)
notes added:
vd1:
  1.  generated by venndiag.ado on 18 Nov 1998 21:48 . Variables and values were
  2.  asthma:1 season:1
  3.  vd1=miss for 1000 records excluded by:in 1/3000
  4.  vd1=miss for 52 records with missing values (.)
```

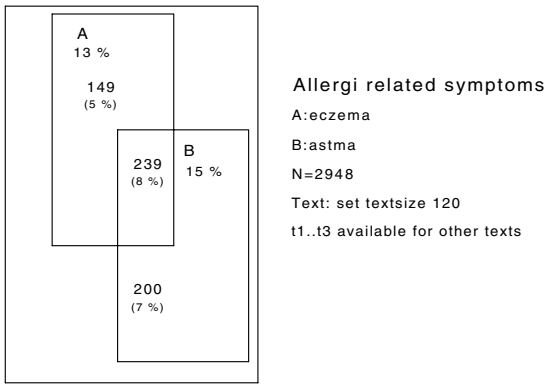


Figure 3. Using a larger text size.

The label of the generated variable will indicate which variables were used for creation, such as

```
. tab vd1
asthma      |
season(vd)  |      Freq.     Percent        Cum.
------------+-----------------------------------
         -- |        2360       59.00       59.00
          A |         149        3.72       62.72
         AB |         239        5.98       68.70
          B |         200        5.00       73.70
       miss |        1052       26.30      100.00
------------+-----------------------------------
      Total |        4000      100.00
```

`A` is the first variable in the label, `B` the second, and so on. When tabulating, the variable records omitted from the graph will be given the value "miss". Here the exclusion was caused by an `in 1/3000` (leaving out 1,000 records) clause and 52 records having missing values in `asthma` and/or `season` variables. This information is saved with the dataset in a note as shown above and can be shown with the general command `note`.

Historical note

John Venn (1834–1923) was British. He worked at the University of Cambridge on logic and developed the "Venn Diagram"—a diagrammatic method of illustrating propositions by inclusive and exclusive circles probably during the years 1866–1881. According to the "Dictionary of National Biography (1922–1930)", page 869, the idea had been developed previous to the publication of *Symbolic Logic* (Venn 1881), but no primary source is given. In the foreword of his book, John Venn states that some of the ideas has been presented earlier and in a historical chapter he writes on page 511: "*So far as I have been able to ascertain, this plan (applied to closed figures) was first employed by Thomson in the second edition of his Laws of Thought in 1849.*" The actual introduction of closed circles to represent combinations of variables therefore most likely dates back to around 1850, but with John Venn deriving more strictly the relationship between logical statements and diagrammatic representations. In the original drawings, Venn applied circles to represent two and three variables and ellipses to represent four variables. (For reasons of programmatic simplicity, squares have been chosen in `venndiag`; future enhancements could change this.)

Acknowledgment

Thanks to Ph. D. M. D. Charlotte G. Mörtz for testing and comments.

Reference

Venn, J. 1881. *Symbolic Logic*. London: Macmillan.

sbe25	Two methods for assessing the goodness-of-fit of age-specific reference intervals

Eileen M. Wright, Imperial College School of Medicine, UK, ewright@rpms.ac.uk
Patrick Royston, Imperial College School of Medicine, UK, proyston@rpms.ac.uk

Many methods have been proposed for the estimation of age-specific reference intervals, which are essentially covariate-specific quantile curves. Parametric models for normally distributed data and for normal or nonnormal data have been implemented in Stata in `xrigls` (Wright and Royston 1997) and `xriml` (Wright and Royston 1996), respectively. Few techniques have been developed to assess the fit of the models. Parametric models generate so-called Z scores, which are standardized residuals. If the model is correctly specified, the Z scores have approximately a standard normal distribution. The first four moments (the mean, variance, skewness and kurtosis) are the expected values of Z, Z^2, Z^3 and Z^4 and are approximately equal to $0, 1, 0$ and 3 respectively, independent of age. An incorrectly specified model may produce nonrandom, age-related patterns in one or more of these moments. The commands presented here, `xriqtest` and `permband`, are intended to detect such patterns.

We introduce two methods of assessing goodness-of-fit.

- `xriqtest` performs five hypothesis tests ("Q tests"), for each of which the null hypothesis is that the Z scores have a standard normal distribution. The observations are divided into several contiguous age groups of roughly equal size. Q tests numbered 1 to 4 are χ^2 tests which compare the first four moments of the Z scores with their (presumed) expected values within each age group. Large values of the test statistics indicate departure from the model. Q_5 uses the Shapiro–Wilk W test (see [R] **swilk**) to test for general departures from a normal distribution within the age groups.

- `permband` is primarily a graphical technique to assess the amount of random variation in the moments of the Z scores that could be expected if there were no age-specific patterns present, i.e., if the original model were correct. The idea is to model local trends in the moments nonparametrically using the running line smooth `running`, of Sasieni and Royston (1998). `permband` plots the smooth of powers of the Z scores together with a "permutation band" based on smooths of the powers

for each of 50 random permutations. For further details, see Royston and Wright (1998a). Evidence of an incorrect model may be present if a smooth crosses the boundaries of its permutation band. More formally, a hypothesis test is performed by comparing the proportion of observations which lie outside the permutation band with a critical value assuming the original model to be correct.

We recommend applying both `xriqtest` and `permband` to the Z scores from models under investigation.

Syntax of xriqtest

```
xriqtest zvar xvar [if exp] [in range] , params(m:#, s:#, g:#) [ groups(# | groupvar)
        mingroup(#) generate ]
```

where *zvar* denotes the Z scores from a model and *xvar* the age variable.

Options for xriqtest

`params(m:#, s:#, g:#)` is not optional and specifies the dimensions (numbers of parameters) of different parts of the model used to estimate age-specific reference intervals. The regression constant for each curve counts as one parameter. The minimal requirement is to specify `params(m:#, s:#)` which specifies that the model is normal with mean and standard deviation curves having the specified number of parameters. The numbers of parameters for `m`, `s`, and `g` are important because they are used to determine the degrees of freedom for the Q tests. For the more general exponential-normal model which may be fitted by `xriml` (see *Technical note*), `m` denotes μ_T, the median curve, `s` denotes σ_T, the scale-factor curve and `g` denotes γ_T, the skewness parameter curve. (Note the effect of fitting a modulus exponential-normal model (`dist(men)` in `xriml`), a further extension which has a parameter representing nonnormal kurtosis, has not been explored.) If, for example, a normal model has been fitted with a quadratic polynomial (or a two-term fractional polynomial) for μ_T and a straight line for σ_T, one would specify `par(m:3,s:2)` (or `par(m:3,s:2,g:0)` which is equivalent). Note that it is the user's responsibility to specify the numbers of parameters correctly as `xriqtest` has no way of checking them; incorrect values will result in misleading test results.

`groups(# | groupvar)` specifies either #, the number of equal-sized groups to be created according to the values of *xvar*, or *groupvar*, an existing variable used to define the groups. If the `groups` option is omitted, a default choice for # is made as follows. Let n be the sample size and let k be the smallest acceptable group size. Then

$$\# = \begin{cases} 1, & \text{if } n < 2 \times k \\ \left[\frac{n}{k}\right], & \text{if } 2 \times k \le n \le 10 \times k \text{ (where [.] means integer part)} \\ 10, & \text{if } n > 10 \times k \end{cases}$$

This choice of # ensures each group is never smaller than k. The value of k is set by using the `mingroup` option. If specified, # must be at least 1 and no more than 50.

`mingroup(#)` determines the smallest acceptable group size (see the `groups` option). The default # is `50`.

`generate` creates 6 new variables as follows:

Variable name	Contents
`_group`	values of grouping variable (1, 2, . . .)
`_mean`	means of grouped Z scores
`_sd`	standard deviations of grouped Z scores
`_pskew`	p values for skewness coefficients
`_pkurt`	p values for kurtosis coefficients
`_pswilk`	p values for Shapiro–Wilk W statistics

Variables with the above names are automatically replaced when the `generate` option is used.

Saved results

`xriqtest` stores in the `S_#` macros:

`S_1`	number of observations	`S_6`	p value for Q_2 test
`S_2`	number of *xvar*	`S_7`	p value for Q_3 test
`S_3`	0± $S_3 is approx. $S_level% confidence interval for means	`S_8`	p value for Q_4 test
`S_4`	1± $S_4 is approx. $S_level% confidence interval for SDs	`S_9`	p value for Q_5 test
`S_5`	p value for Q_1 test		

Syntax for permband

```
permband  zvar [xvar] [if exp] [in range] , [ alpha(#) moment(#) reps(#) seed(#) span(#)
          nograph graph_options ]
```

zvar denotes the Z scores from a model and *xvar* the age variable. If *xvar* is not supplied it is taken to be the order number (1,2,3, ...) of the nonmissing observations.

Options for permband

`alpha(#)` sets the significance level for the hypothesis test of the proportion of values outside the permutation band. The default value is 0.05.

`moment(#)` specifies which moment (1, 2, 3 or 4) of the Z scores is to be investigated. The hypothesis test is not performed for the fourth moment. The default value of # is 1, meaning the first moment which relates to the mean curve (μ_T) in the original model.

`reps(#)` sets the number of permutations. The hypothesis test is given only for 50 permutations. The default value of # is 50.

`seed(#)` defines the random number seed for selecting random permutations of *zvar*. The default is the current system seed.

`span(#)` determines the width of the smoothing window that is applied to *zvar*. The default is as set by the `running` command with the `double` option, namely $2\left[n^{0.8}/2\sqrt{2}+1/2\right]/n$.

`nograph` specifies no graph is to be produced.

graph_options are any of Stata's `graph, twoway` options except `sort`.

Example

Figure 1 shows observations of fetal humerus length from 613 pregnancies in relation to gestational age, together with a 90% age-specific reference interval estimated using `xriml`.

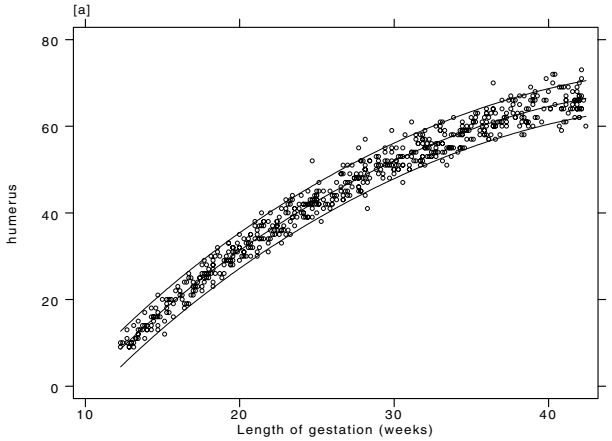


Figure 1. 5th, 50th and 95th centiles for fetal humerus length vs gestational age.

Figure 2 shows the Z scores from the model.

(*Graph on next page*)

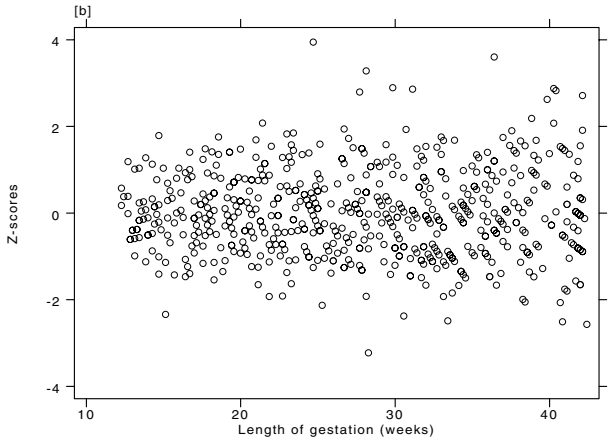


Figure 2. Corresponding Z-scores vs gestational age.

The model assumes the data are normally distributed. A fractional polynomial (see [R] **fracpoly**) with powers $(0.5, 3)$ in gestational age was chosen for the age-specific mean, and a constant for the standard deviation. For the normal model, the Z scores (see Figure 2) are equal to (humerus length minus age-specific mean)$\big/$(standard deviation).

Consider again the plot of Z scores (`Z`) vs gestational age (`gest`) in Figure 2. Since the values seem to "fan out," there may be some heteroscedasticity in the Z scores. To investigate the model fit more formally, we apply `xriqtest`. The output is shown below.

```
. xriqtest Z gest, par(m:3,s:1) generate
Q-tests on Z-scores in Z with 10 groups by gest (N = 613):
 Null hypothesis        Test      Chi-sq      DF      P-value
 ------------------------------------------------------------
 Means = 0               Q_1      5.7454       7        0.570
 Variances = 1           Q_2      26.599       9        0.002
 Skewness coeffs = 0     Q_3      17.289      10        0.068
 Kurtosis coeffs = 3     Q_4      19.472      10        0.035
 Normality               Q_5      30.934      20        0.056
```

The results confirm the subjective impression from Figure 1. The `generate` option creates the grouping variable and variables containing age-group-related statistics. Figure 3 shows the within-age-group standard deviations of the Z scores (`_sd`) plotted against age group (`_group`).

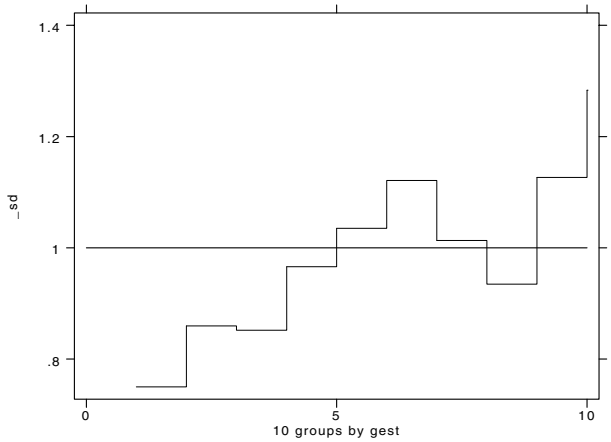


Figure 3. `_sd` variable from Q_2 statistic vs `_group`, both outputted from `xriqtest`.

Overall, the standard deviation increases with age group. It is helpful to see also a smoothed plot of the relationship and to quantify the amount of variation expected by chance. Such information is gained when `permband` is applied to the second moment of `Z`. Figure 4 shows the resulting smooth of $\texttt{Z}^2$ and its permutation band. The output for the corresponding test is given below. Note that `xlabel`, `ylabel`, `border` and `gap(5)` are simply graph options.

```
. permband Z gest, mom(2) seed(125) xlabel ylabel border gap(5)
.............................................
  Variable |   Obs       Proportion         Critical value      P-value
           |             outside band        (alpha = .05)
 ----------+-----------------------------------------------------------
       Z^2 |   613          .3866              .1036              0.000
```

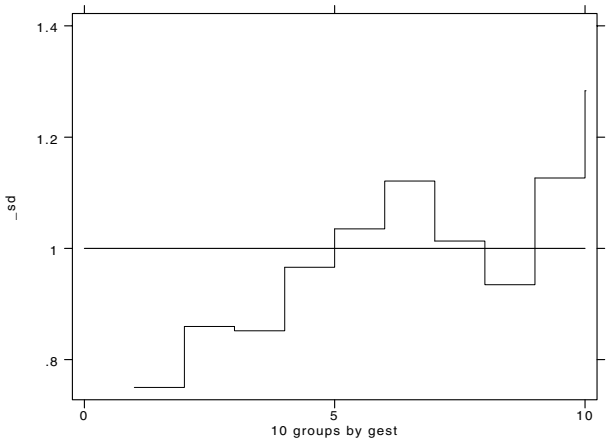


Figure 4. Permutation band for Z^2

The test has a highly significant p value, confirming the presence of age-specific trends in the second moment. The trend in the smoothed Z^2 values of Figure 3 is similar to that shown in Figure 2 from `xriqtest`. If we wished, we could generate and plot the square root of the smooth (`s2_Z`) of Z^2 and the permutation band boundaries (`l2_Z`, `u2_Z`) to view the shape of the curves in terms of the standard deviation rather than the variance.

To improve the model, a linear term in gestational age is included for the standard deviation. The new model is fitted using `xriml ( xriml humerus gest, fp(m:0.5 3,s:1) dist(n))`. The results of the Q tests for the resulting Z scores (`Znew`) are given below.

```
. xriqtest Znew gest, par(m:3,s:2)
Q-tests on Z-scores in Znew with 10 groups by gest (N = 613):
 Null hypothesis        Test      Chi-sq      DF      P-value
 ------------------------------------------------------------
 Means = 0               Q_1      5.2513       7        0.629
 Variances = 1           Q_2      6.2157       9        0.718
 Skewness coeffs = 0     Q_3      16.767      10        0.080
 Kurtosis coeffs = 3     Q_4      18.841      10        0.042
 Normality               Q_5       31.05      20        0.055
```

The standard deviation now appears to be adequately modeled. The results, however, suggest nonnormality in `Znew`. A permutation band for the third moment was calculated (see Figure 5 and output below).

```
. permband Znew gest, mom(3) seed(129) yline(0) xla yla bord gap(5)
.............................................
  Variable |   Obs       Proportion         Critical value      P-value
           |             outside band        (alpha = .05)
 ----------+-----------------------------------------------------------
    Znew^3 |   613             0             .1464              1.000
```

(*Graph on next page*)

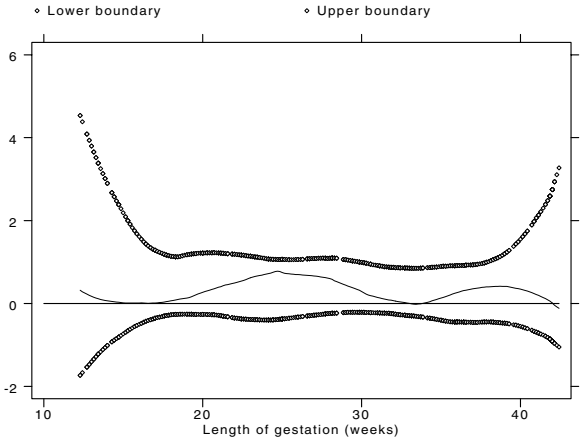


Figure 5. Permutation band for `Znew`[3]

Figure 5 indicates that no simple age-specific trends in `Znew`[3] are present, but the smooth (`s3_Znew`) does appear to be slightly above the expected value of zero at all gestational ages, suggesting skewness. The simplest way to accommodate skewness is to fit an EN model with constant γ_T (i.e. `xriml humerus gest, fp(m:0.5 3,s:1) dist(en)`). The result is a significant improvement in fit compared with the normal model ($P = 0.003$, likelihood-ratio test). The Z scores from the EN model were examined. All the p values for the Q tests are > 0.1 and the smooths, permutation bands and tests for the first 3 moments from `permband` show no evidence of age-specific trends. We conclude that the model appears to fit adequately.

If evidence of nonnormal kurtosis was found in the `EN` model, e.g., a significant p value for the Q_4 test of the Z scores, a modulus exponential-normal (MEN) model may be fitted using `dist(men)` with `xriml`. A subjective impression of age-specific trends in kurtosis may be gained from `permband` using the option `mom(4)` although a corresponding test has not been developed.

Technical note

Royston and Wright (1998b) describe parametric models based on the normal and exponential-normal distributions to estimate age-specific reference intervals. Z scores for the normal distribution are defined by $Z = (Y - \mu_T)/\sigma_T$, where μ_T and σ_T are respectively the mean and standard deviation of $Y|T$. Z scores for exponential-normal distribution are defined by $Z = \dfrac{1}{\gamma_T}\left\{\exp\left(\gamma_T\left(Y - \mu_T\right)/\sigma_T\right) - 1\right\}$ where μ_T and σ_T are respectively the median and scale parameter of $Y|T$, and $-1/3\gamma_T$ approximately equals the skewness (standardized third moment) of $Y|T$. Royston and Wright (1998b) also describe the modulus-exponential normal model which includes a fourth parameter relating to kurtosis. Goodness-of-fit for these models was not considered by Royston and Wright (1998a).

Suppose we have $n \times 1$ observation vectors (t, y) and Z scores z obtained after fitting a model to (t, y). For example, y may be observations of fetal humerus length and t the corresponding gestational ages.

Procedure for xriqtest

The procedure used by `xriqtest` is as follows (see Royston and Wright, 1998a for further information):

1. The elements of z are divided into G contiguous subsets of sizes $n_1, \ldots, n_G$.

2. The group means $\overline{z}_1, \ldots, \overline{z}_G$ of z are compared with their expected values of zero using the statistic

$$Q_1 = \sum_{g=1}^{G} n_g \overline{z}_g^2$$

   which has a χ^2 distribution on $G - \dim(\mu_T)$ degrees of freedom, where $\dim(\mu_T)$ denotes the number of estimated parameters in the model for μ_T.

3. The cube roots of the sample variances $v_1, \ldots, v_G$ of z are compared with their expected values of $1 - \dfrac{2}{9(n_g - 1)}$ using the statistic

$$Q_2 = \left\{ \sum_{g=1}^{G} v_g^{1/3} - \left(1 - \frac{2}{9\,(n_g - 1)} \right) \right\}^2 \Bigg/ \frac{2}{9\,(n_g - 1)}$$

Q_2 has approximately a χ^2 distribution on $G - \left[\frac{1}{2} \dim(\sigma_T) + \frac{1}{2} \right]$ degrees of freedom.

4. Let $p_1, \ldots, p_G$ be the two-tailed p values from a test of zero skewness (see [R] **sktest**) applied within each group. The statistic

$$Q_3 = \sum_{g=1}^{G} \left[\Phi^{-1} \left(\frac{1}{2} p_g \right) \right]^2$$

where Φ^{-1} is the standard normal quantile function, is sensitive to small (i.e., significant) values of $p_1, \ldots, p_G$ which indicate nonnormal skewness. Q_3 has approximately a χ^2 distribution on $G - \dim(\gamma_T)$ degrees of freedom.

5. A similar computation is performed for the p values from a test of normal kurtosis (see [R] **sktest**) applied within each group, giving a statistic Q_4 also with approximately a χ^2 distribution on G degrees of freedom.

6. The p values $p_1, \ldots, p_G$ from the Shapiro–Wilk W test of nonnormality (see [R] **swilk**) applied within each group are combined using Fisher's method, giving the test statistic

$$Q_5 = \sum_{g=1}^{G} -2 \ln(p_g)$$

which has a χ^2 distribution on $2G - \dim(\gamma_T)$ degrees of freedom.

The tests have been validated with sample sizes $50 \leq n \leq 1000$. They are expected to be valid for $n > 1000$, though possibly not for $n < 50$.

Procedure for permband

The procedure used by `permband` is related to the idea of envelopes for normal plots (Atkinson 1985) and may be written as follows (see Royston and Wright 1998a for further information):

1. Keeping t fixed, create 50 element-wise random permutations $z_1, \ldots, z_{50}$ of z.

2. Using `running` (Sasieni and Royston 1998) with the `double` option to enhance smoothness, compute scatterplot smooths $s_0, s_1, \ldots, s_{50}$ of $z, z_1, \ldots, z_{50}$ with respect to t.

3. Create $s_{\min}$ and $s_{\max}$ to be $n \times 1$ vectors containing the observation-wise minimum and maximum of $s_1, \ldots, s_{50}$.

4. To reduce the jaggedness of $s_{\min}$ and $s_{\max}$, use `running` (without the `double` option) to calculate smooths $\tilde{s}_{\min}$ and $\tilde{s}_{\max}$ of $s_{\min}$ and $s_{\max}$ with respect to t. $(\tilde{s}_{\min}, \tilde{s}_{\max})$ defines the lower and upper boundaries of the permutation band for s_0.

5. Plot s_0, $\tilde{s}_{\min}$, and $\tilde{s}_{\max}$ against t and calculate the proportion B_1 of observations for which s_0 lies outside the permutation band, that is,

$$B_1 = n^{-1} \left\{ \sum_{i=1}^{n} I\left(s_{0;i} < \tilde{s}_{\min;i}\right) + I\left(s_{0;i} > \tilde{s}_{\max;i}\right) \right\}$$

where $I(\cdot)$ is an indicator function.

The shape of s_0 may show where the model for the mean curve μ_T is misspecified. The quantity B_1 is a nonparametric test statistic for T dependence of $E(Z)$, a large value suggesting nonrandom behavior and therefore that μ_T is misspecified.

The procedures for $j = 2, 3$ are identical, except that z is replaced with z^j and B_1 with B_j. The interpretations of the plots and of B_2 and B_3 relate to the models for σ_T and γ_T respectively. We have not found plots of age-specific kurtosis of Z scores (i.e. of z^4) to be useful. The distribution of Z^4 is exceptionally long-tailed and the amount of information in the smooths of z^4 seems to be too small to be meaningful.

Acknowledgments

This research received financial support from project grant number 039911/Z/93/Z from The Wellcome Trust. We are grateful to Lyn Chitty for allowing the use of her data.

References

Atkinson, A. C. 1985. Plots, transformations and regression. *Oxford Statistical Science Series* 1: 34–37. Oxford: Oxford University Press.

Royston, P. and E. M. Wright. 1998a. Goodness-of-fit statistics for age-specific reference intervals. *Statistics in Medicine*, submitted.

——. 1998b. A method for estimating age-specific reference intervals ("normal ranges") based on fractional polynomials and exponential transformation. *Journal of the Royal Statistical Society*, Series A 161: 79–101.

Sasieni, P. and P. Royston. 1998. sed9.1: Pointwise confidence intervals for running. *Stata Technical Bulletin* 41: 17–23. Reprinted in *Stata Technical Bulletin Reprints*, vol. 7, pp. 156–163.

Wright, E. M. and P. Royston. 1996. seb13: Age-specific reference intervals ("normal ranges"). *Stata Technical Bulletin* 34: 24–34. Reprinted in *Stata Technical Bulletin Reprints*, vol. 6, pp. 91–104.

——. 1997. seb15: Age-specific reference intervals for normally distributed data. *Stata Technical Bulletin* 38: 4–9. Reprinted in *Stata Technical Bulletin Reprints*, vol. 7, pp. 93–100.

sbe26 | Assessing the influence of a single study in the meta-analysis estimate

Aurelio Tobías, Institut Municipal d'Investigacio Medica (IMIM), Barcelona, atobias@imim.es

Graphical methods are regularly used in meta-analysis to complement the statistical analysis of epidemiological and clinical data. At the moment the following graphical methods to complement the meta-analysis results are available in Stata: the Galbraith (Tobas 1998) and L'Abbe plots (Bradburn et al. 1998) to investigate heterogeneity, however we can also test if one or more covariates with values defined for each study in the analysis explain heterogeneity doing a meta-analysis regression (Sharp 1998), the funnel (Steichen et al. 1998, Bradburn et al. 1998) and Egger plots (Steichen et al. 1998) to check for publication bias, and finally we can track the accumulation of evidence of the effect doing a cumulative meta-analysis (Sterne 1998). Now I present a graphical technique to look for influential studies in the meta-analysis estimate.

The `metainf` command investigates the influence of a single study on the overall meta-analysis estimate. This command shows graphically, as a standard error bar chart, the results of an influence analysis, in which the meta-analysis estimates are computed omitting one study in each turn. This technique has been frequently used in meta-analyses of epidemiologic studies (see, for example, Kogevinas et al. 1997 or Lubin et al. 1997).

Syntax

As other commands related to meta-analysis, the command `metainf` works on a dataset containing the estimate effect, *theta*, and its standard error, *setheta*, for each trial. The syntax is as follows:

`metainf` *theta setheta* [`if` *exp*] [`in` *range*] [`, id(`*labvar*`) eform random print` *graph_options*]

Options

`id(`*labvar*`)` is a variable which is used to label the studies.

`eform` requests that the output is exponentiated.

`random` request random-effects estimates. By default fixed-effects estimates are computed.

`print` shows the estimates, and their 95% confidence intervals, of the meta-analysis estimates omitting one study in each turn.

graph_options are any of the options allowed with `graph, twoway` except `yline()`.

Example

I will illustrate the use of the `metainf` command with data from eight epidemiologic studies on the lung cancer risk from residential radon (Lubin et al. 1997). They checked if there were any influential study in the overall meta-analysis estimate.

```
. describe
Contains data from lubin97.dta
obs:             8
 vars:           4
 size:           200
-------------------------------------------------------------------------------
   1. study       float  %9.0g                 study identify number
   2. studyid     str9   %9s                   study identify label
   3. logrr       float  %9.2f                 log relative risk
   4. logse       float  %9.2f                 standard error log relative ris
-------------------------------------------------------------------------------
```

```
. list, noobs
    study    studyid      logrr      logse
        1   Finland1       0.26       0.09
        2   Finland2       0.01       0.04
        3  NewJersey       0.60       0.23
        4   Shenyang      -0.17       0.04
        5   Winnipeg      -0.04       0.06
        6  Stockholm       0.60       0.16
        7     Sweden       0.18       0.03
        8   Missouri       0.11       0.10
```

Heterogeneity was tested using the `meta`-command (Sharp and Sterne 1998), with strong statistical evidence of its presence ($Q = 77.709$, $df = 7$, $p < 0.001$). A random effects models should be used, as the authors suggested. However, we can also check if there is any influential study in the overall random-effects estimate. We can see from Figure 1 that the overall estimates changed very little when any single study is omitted.

```
.  meta logrr logse, eform
Meta-analysis (exponential form)
       |  Pooled      95% CI         Asymptotic      No. of
Method |     Est   Lower   Upper  z_value  p_value   studies
-------+-----------------------------------------------------
Fixed  |   1.064   1.027   1.103    3.415    0.001       8
Random |   1.143   0.994   1.313    1.881    0.060
Test for heterogeneity: Q= 77.709 on 7 degrees of freedom (p= 0.000)
Moment-based estimate of between studies variance =  0.032

. set textsize 75
. metainf logrr logse, id(studyid) random eform print
Meta-analysis Random-effects estimates
------------------------------------------------------------------------------
 Study omitted     |   e^coef.        [95%  Conf.  Interval]
-------------------+----------------------------------------------------------
 Finland1          |   1.1219499      .96613681    1.3028917
 Finland2          |   1.1828868      .99321449    1.4087806
 NewJersey         |   1.1490161      .98619378    1.3387208
 Shenyang          |   1.1088951      .96392721    1.275665
 Winnipeg          |   1.1873517      1.0501852    1.3424337
 Stockholm         |   1.0882355      .94881761    1.2481393
 Sweden            |   1.1340759      .97280651    1.3220801
 Missouri          |   1.182092       1.0089555    1.3849387
-------------------+----------------------------------------------------------
 Combined          |   1.1428653      .99441699    1.3134742
------------------------------------------------------------------------------
```

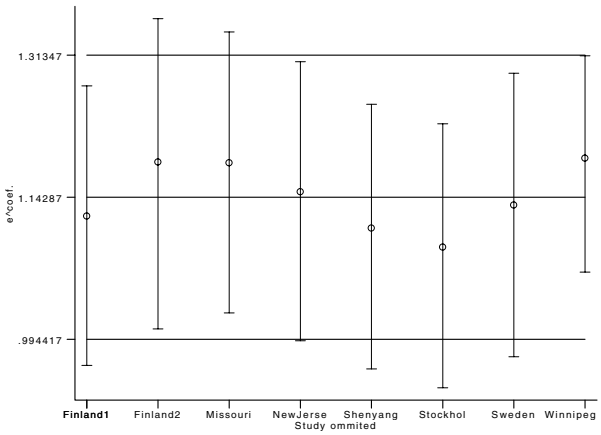


Figure 1. The influence graph for the meta-analysis of eight studies evaluated.

Individual or frequency records

As in other meta-analysis commands in Stata, `metainf` works on data contained in frequency records, one for each study. If we have primary data, that is, individual records, we must combine into frequency records using the `collapse` and `by var` Stata commands.

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.

Kogevinas, M., H. Becher, T. Been, et al. 1997. Cancer mortality in workers exposed to phenoxy herbicides chlorophenols, and dioxins. *American Journal of Epidemiology* 145: 1061–1075.

Lubin, J. H. and J. D. Boice. 1997. Lung cancer risk and residential radon: meta-analysis of eight epidemiological studies. *Journal of the National Cancer Institute* 89: 49–57.

Sharp, S. 1998. sbe23: Meta-analysis regression. *Stata Technical Bulletin* 42: 16–22. Reprinted in *The Stata Technical Bulletin Reprints*, vol. 7, pp. 148–155.

Sharp, S. and J. Sterne. 1998. sbe16.1: New syntax and output for the meta-analysis command. *Stata Technical Bulletin* 42: 6–8. Reprinted in *The Stata Technical Bulletin Reprints*, vol. 7, pp. 106–108.

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

Sterne, J. 1998. sbe22: Cumulative meta-analysis. *Stata Technical Bulletin* 42: 13–16. Reprinted in *The Stata Technical Bulletin Reprints*, vol. 7, pp. 143–147.

Tobas, A. 1998. sbe20: Assessing heterogeneity in meta-analysis. *Stata Technical Bulletin* 41: 15–17. Reprinted in *The Stata Technical Bulletin Reprints*, vol. 7, pp. 133–136.

sg99	Multiple regression with missing observations for some variables

Mead Over, World Bank, meadover@worldbank.org

Missing values of important variables is a frequently encountered frustration in applied econometrics and biometrics. The analyst typically chooses between estimating the full model on a small number of observations or dropping variables from the model in order to estimate it on a larger number of observations. The third alternative, imputing the values of some of the missing observations in order to retain all the theoretically relevant variables with a larger sample, is a riskier strategy, which is tedious to implement and document. The Stata command `regmsng` presented here is designed to help an analyst systematically apply and document this third alternative. The `regmsng` command performs ordinary least squares multiple regression on a dataset that includes missing values of some of the right-hand-side variables. The command implements any of several current imputation procedures, but makes no attempt to correct the estimated standard errors of the coefficients for the fact that some of the data has been imputed.

Replacing values that are "ignorably missing"

Standard textbook treatments of the subject of missing values divide the problem into several distinct cases (Greene 1997, 427–32). The first major distinction is between missing values on the left-hand side (i.e., in the dependent variable) and missing values on the right-hand side (i.e., among the independent variables). When values of the left-hand side variable are missing, analysts agree that imputing their values in order to expand the sample is likely to be dangerous except in very special circumstances. This note and the `regmsng` command do not address the possibility of missing values on the left-hand side of a regression equation. Attention here is restricted to the case where there are missing values among the right-hand-side variables.

A second distinction is between so-called "ignorably missing" data on the right-hand side and situations in which the missing values are not "ignorable." The term "ignorably missing" was first applied by Griliches (1986) to describe the situation in which the coefficients of a regression equation can be estimated consistently on the subset of the data that includes no missing values. This will be the case when the presence of a value for a given observation of a right-hand-side variable is independent of the error term of the regression being estimated. On the other hand, if observations with particularly large positive stochastic disturbances (for example) are more likely to be missing some values of the right-hand side variables, then estimating the regression on the subset of data which contains no missing values is likely to produce biased or inconsistent estimates of the coefficients.

In this note we adopt the assumption that the "holes" in a dataset consist only of ignorably missing values. We assume that there is no systematic relationship between the regression's disturbance term and the presence or absence of a value on any of the right-hand-side variables. This ignorably missing case is the one in which there is the most hope of expanding the number of observations in a regression by imputing some of the missing values without introducing inconsistency or reducing efficiency. At the end of this note, we briefly address the question of how to tell whether the missing data is ignorable.

Since in the ignorably missing case estimation of a regression on only the subset of complete observations is (by assumption) guaranteed to produce consistent estimates of the coefficients, the only reason to attempt to impute missing values would be to improve the efficiency of the estimates. Whether the efficiency gains from imputing some values of missing right-hand-side variables will be worth the additional unknown error introduced by the imputation process will be a matter of judgment in any individual case. The situation in which the promise of improved efficiency seems particularly tempting is when there are observations which are "almost complete." That is, the model includes $k > 1$ right-hand-side variables and could be estimated on N_c complete observations. However, suppose there are another N_1 observations with data on $k - 1$ of the right-hand-side variables, N_2 with data on $k - 2$, and so on. If N_1 is large, especially in relation to N_c, it is painful to omit those additional

almost complete observations from the regression knowing that 1) by imputing one variable on each observation one adds good data from $k-1$ variables to the regression, and 2) demonstration that the regression model is robust to the inclusion of the extra N_1 observations would lend credence to the hypothesis that the missing variables are ignorably so. The same argument applies, though with less force, to the addition of N_2 observations missing on only 2 of the k variables. This is the situation that `regmsng` is specifically designed to address.

Syntax

`regmsng` *yvar xvarlist* [*weight*] [`if` *exp*] , `nmiss(#)` [`fitted` `inst(`*varlist*`)` `steps` `change` `nohist` `notest` `noaux` `level(#)`]

All four types of weights are accepted and applied in the intermediate calculations producing the means or the fitted values as well as in the final regression. The variable or variables that define the weight are assumed not to have any missing values. Use weights with caution.

Issuing the `regmsng` command without a variable list or options causes Stata to display the results of the last execution of `regmsng`.

`regmsng` has the following restrictions:

1. The command is intended to apply only to multiple regression. Therefore, there must be at least two right-hand-side variables.
2. The names of the right-hand-side variables must have no more than seven characters (to allow creation of a new variable called `Ax` for any right-hand-side variable `x` with missing values). Furthermore, neither the left- nor the right-hand-side variables can begin with a capital `A`.

Options

`nmiss(#)` specifies that observations for which more than this number of right-hand-side variables have missing data will be dropped from the regression. `nmiss` is a required option.

`fitted` means to replace missing values with the fitted values from regressions, a technique sometimes referred to as "first-order mean replacement". Candidates for regressors in an auxiliary regression to predict missing values of a right-hand-side variable `x` include the other right-hand-side variables in the original regression plus the instrumental variables specified in the `inst` option, if any. Candidate variables are rejected, however, if they are missing on any of the observations on which `x` is missing. If `fitted` is not specified, the default is to replace the missing values by the means of those values (i.e., "zero-order mean replacement").

`inst(`*varlist)* can only be specified with `fitted` and specifies additional instrumental variables to use in predicting the missing values of the right-hand-side variables, i.e., variables not included among the list of right-hand-side variables for the regression. For predicting any given right-hand-side variable, the only variables used are those other right-hand-side variables or instruments which are not missing on all the observations where the given variable is missing.

`steps` causes the program to document the steps involved in imputing the missing values. If `fitted` was chosen, the first-stage regressions are presented. Otherwise the variable means are presented.

`change` allows the program to leave behind changed data and the new variables. It is important to realize that this option destroys the data in memory before the command was issued. Use this option carefully.

`nohist` suppresses the output of the tabulation showing the number of observations missing 0, 1, 2, and so on values.

`notest` suppresses the output of the tests that otherwise follow the estimation.

`noaux` substitutes imputed values for missing values on the right-hand side. The default (if this option is not specified), is to define an auxiliary variable to contain the imputed values and to replace the missing values of the right-hand-side variable with zeros.

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

A few simple examples

The command

```
. regmsng y x1 x2 x3, nmiss(1)
```

estimates a regression of y on x1, x2, and x3 including not only the observations with no missing values, but also those that are complete except for one of the three independent variables. In other words, drop observations that are missing more than one of the three right-hand-side variables. Missing values are replaced by means, which are used to construct auxiliary variables Ax1, Ax2, and Ax3.

The command

```
. regmsng y x1 x2 x3, nmiss(2) fitted inst(x4 t)
```

loosens the restriction to include observations that are missing either one or two of the three right-hand-side variables. It estimates missing values by the fitted values of regressions on the other right-hand-side variables plus two other exogenous variables, x4 and a date or time trend variable called t.

The command

```
. regmsng y x1 x2 x3, nmiss(2) fitted inst(x4 t) nohist notest
```

is the same as the previous command but suppresses some of the output.

Finally,

```
. regmsng y x1 x2 x3, nmiss(2) fitted inst(x4 t) steps
```

is the same as above, but giving complete output.

A detailed example

In order to provide a concrete example of the application of regmsng, this note applies the command to a cross-country dataset which was constructed in order to explore the socioeconomic determinants of the level of HIV infection in the urban areas of developing countries circa 1994. The dataset is called aidsdata.dta and is provided on the STB diskette.

```
. use aidsdata, clear
(X-cntry data on HIV & Dtrmnts)
. describe
Contains data from aidsdata.dta
  obs:           179                          X-cntry data on HIV & Dtrmnts
 vars:            17                          26 Jul 1998 07:43
 size:        17,184 (98.0% of memory free)
-------------------------------------------------------------------------------
   1. cntry       str24  %24s                 Country Name
   2. cn          int    %8.0g       cn        Country ID
   3. region      float  %9.0g       reg2      Region of the World
   4. logitP      float  %9.0g                 Logit of Urban HIV Prev. Rate
   5. hirisk      byte   %8.0g                 Dummy = 1 for High Risk Group
   6. epiage      byte   %8.0g                 Age of the epidemic
   7. lgnppci     float  %9.0g                 Log of 1990 Intnl GNP PC
   8. gini        double %9.3f                 1980s GINI Coefficient
   9. urbmf       double %9.3f                 1990 Urban Male Female Ratio
  10. miltpop     float  %9.0g                 Soldiers per 1000 urban pop
  11. gaplit      float  %9.0g                 1985 Literacy Gap: M - F
  12. muslim      double %12.0g                Percentage of Pop Muslim
  13. lmstp90     float  %9.0g                 Log of 1990 % Frgn Born
  14. lifex       float  %9.3f                 1994 Life Expectancy
  15. imrt80      float  %12.0g                Infant mortality rate, 1980
  16. lgnppc2     float  %9.0g                 Square of Log(GNP/POP)
  17. lpop        float  %9.0g                 Log of Population
-------------------------------------------------------------------------------
Sorted by:  cn
. summarize
Variable |     Obs        Mean   Std. Dev.       Min        Max
---------+-----------------------------------------------------
   cntry |       0
      cn |     179    100.3017   56.17277          3        197
  region |     179    2.787709   1.951567          1          8
  logitP |     179   -2.927862   2.347326  -6.801283   3.068053
  hirisk |     179    .4636872   .5000785          0          1
  epiage |     175    10.20571   2.895294          0         13
 lgnppci |     164    7.819275   .9286142   6.040255   9.983685
    gini |     118    .4459212   .0867954       .279        .62
   urbmf |     144    1.091489   .2068712   .6631016   1.673077
 miltpop |     153    11.54701   12.73505          0    72.0306
```

```
    gaplit |     143    15.18182   10.65655  -21.79998   30.20002
    muslim |     157    22.11338   34.15147          0        100
  lmstp90 |     169     .358626   1.348776  -2.302585   3.407842
    lifex |     167    61.06408   10.16585   38.45888   78.43414
    imrt80 |     164    87.44402   46.15677       11.2      201.2
  lgnppc2 |     164    61.99813   14.77671   36.48468   99.67396
     lpop |     177      8.8108   1.996598   3.367296   13.96819
```

The results of the analysis are presented in World Bank (1997) and a complete discussion of this analysis, including a description of all of the variables, the motivation for their inclusion in the regression, and interpretation of the results, is contained in Over (1998). Suffice it to say here that the dependent variable in the analysis is a specific transformation of the percentage infected with HIV of a specific risk category of urban residents in each country. In 76 countries, observations were present for both "low risk" and "high risk" urban adults. In a further 27 countries, the percent infected (or "prevalence rate") was only available for one or the other of the two risk groups. Pooling the data gives a total of 179 observations. The transformation of the prevalence rate is called `logitP` and is defined as

$$\texttt{logitP} = \ln\left(\frac{P}{C-P}\right)$$

where P is the prevalence rate and C is the assumed ceiling rate for the risk group. The ceiling rates are assumed to be 90 percent for the high risk group and 40 percent for the low-risk group. Also, measured values of P equal to zero are replaced by 0.1% in the low risk group and by 1% in the high risk group.

In this example, the eight socio-economic variables `epiage`, `lgnppci`, `gini`, `urbmf`, `miltpop`, `gaplit`, `muslim`, and `lmstp90` are hypothesized to influence the level of urban HIV prevalence in a given country. Note that none of these independent variables is complete. On one of the variables, `gini`, there are only 118 observations.

Using regmsng to increase the number of usable observations

Like most statistical packages, Stata omits an observation from a regression if any of the left- or right-hand-side variables contains a missing value for that observation. When we run a regression of `logitP` on the eight independent variables, we discover that only 96 observations are complete. If we use the standard conservative approach to estimating the regression and insist on retaining all eight of the independent variables, we must discard 83 of the 179 observations, almost half the data.

`regmsng` is just as finicky as `regress` with respect to the left-hand-side variable, but more tolerant of missing observations on the right-hand side. Let n_m be the maximum number of right-hand-side variables containing a missing value on any observation in a dataset. Standard practice in `regress`, `fit` and other members of Stata's estimation family is to drop the observation if $n_m > 0$. `regmsng` instead allows the user to set n_m to an integer greater than zero using the `nmiss` option. For example, if n_m is set to equal two, an observation would not be dropped unless three or more right-hand-side variables are missing on that observation.

In order to assist the user in selecting a value of n_m, the first output of `regmsng` is a histogram which presents the number of observations on which zero, one, two, three, and so on, are missing. For the sample dataset, here is the histogram produced by `regmsng` (note that `RHS` is a global macro containing the names of the independent variables):

```
. global RHS epiage lgnppci gini urbmf  miltpop gaplit muslim lmstp90
. global INST rgn1 rgn2 rgn3 rgn5 rgn6 rgn7 lifex imrt80 lgnppc2 lpop
. regmsng logitP hirisk $RHS , nmiss(3) steps noaux
Number of observations with 0, 1, 2, ... RHS variables missing
No. of RHS  |
variables   |
missing     |      Freq.
------------+------------+-----------------------------------------------------
          0 |         96 |**************************************************
          1 |         36 |******************
          2 |         14 |********
          3 |         13 |*******
          4 |         10 |******
          5 |          1 |*
          6 |          3 |**
          7 |          5 |***
          8 |          1 |*
------------+------------+-----------------------------------------------------
      Total |        179
  (output omitted )
```

The histogram informs us that 36 additional observations can be added by setting n_m to one, instead of to zero as the conservative approach would dictate. Examination of the data reveals that these observations add 22 countries to a sample that otherwise would consist of only 50 countries. Increasing n_m to two would add an additional 14 observations (8 countries), and so on. In order to include all 179 observations, it would be necessary to set n_m to eight. Note that doing so would add only a single observation more than when n_m is set to seven. Furthermore, this last observation contains data on the dependent variable, but is missing data on all eight of the independent variables.

Intuition suggests that it may well be worthwhile to impute the values of a few independent variables in order to increase n_m from zero to one and add 36 observations. On the other hand, most analysts would be reticent to include observations on which all eight variables would have to be imputed. The difficult question is where in between these two extremes should the line be drawn on adding observations at the expense of introducing errors in imputation. This issue is raised again below, after we discuss the methods that `regmsng` uses to impute missing values and estimate the regression.

The most obvious approach to including observations that contain missing values of some variables is to replace the missing values of the variables with imputed values. `regmsng` takes this approach when the `noaux` option is specified. However, the default approach is to replace the missing values of the variable `x` with zeros and define an auxiliary variable, say `Ax`, which contains zeros where `x` is present and the imputed values where `x` is missing. This alternative allows the regression to estimate separate regression coefficients for the original variable `x` and for the new auxiliary variable `Ax`. If the estimated coefficients of `x` and `Ax` differ, constraining them to be equal with the `noaux` option will damage the efficiency of the estimates of the other coefficients.

Here are the results of estimating the `logitP` regression with `nmiss` equal to three, mean-replacement of the missing variables, and using the `noaux` option:

```
. regmsng logitP hirisk $RHS , nmiss(3) steps noaux nohist
Regression with Imputed Values of Missing R.H.S. Variables
Means substituted for missing observations of RHS variables.
  Source |       SS       df       MS                  Number of obs =     159
---------+------------------------------               F(  9,   149) =   18.93
   Model |  463.945753     9  51.5495281               Prob > F      =  0.0000
Residual |  405.670153   149  2.72261848               R-squared     =  0.5335
---------+------------------------------               Adj R-squared =  0.5053
   Total |  869.615906   158  5.50389814               Root MSE      =    1.65
------------------------------------------------------------------------------
  logitP |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
  hirisk |    1.40347   .2629195      5.338   0.000       .8839377    1.923002
  epiage |   .2465037    .057182      4.311   0.000       .1335113    .3594961
 lgnppci |  -1.049516   .1607009     -6.531   0.000      -1.367063   -.7319689
    gini |   8.608299   1.943875      4.428   0.000       4.767176    12.44942
   urbmf |   2.846628   .8184595      3.478   0.001       1.229342    4.463915
 miltpop |   .0239794   .0125455      1.911   0.058      -.0008107    .0487694
  gaplit |    .027516   .0174325      1.578   0.117      -.0069308    .0619629
  muslim |  -.0172087   .0044527     -3.865   0.000      -.0260073   -.0084101
 lmstp90 |   .2726888   .1073213      2.541   0.012       .0606206     .484757
   _cons |  -5.132751   2.165936     -2.370   0.019      -9.412669   -.8528329
------------------------------------------------------------------------------
```

Next we use the default option and use the auxiliary variables beginning with `A`:

```
. regmsng logitP hirisk $RHS , nmiss(3) nohist notest
Regression with Imputed Values of Missing R.H.S. Variables
A-variables contain the mean values of corresponding independent variables.
  Source |       SS       df       MS                  Number of obs =     159
---------+------------------------------               F( 16,   142) =   10.98
   Model |  480.870831    16   30.054427               Prob > F      =  0.0000
Residual |  388.745074   142  2.73764137               R-squared     =  0.5530
---------+------------------------------               Adj R-squared =  0.5026
   Total |  869.615906   158  5.50389814               Root MSE      =  1.6546
------------------------------------------------------------------------------
  logitP |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
  hirisk |   1.450651   .2666164      5.441   0.000       .9236006    1.977701
  epiage |   .2219185   .0689596      3.218   0.002       .0855985    .3582386
 lgnppci |  -.9215413   .1799452     -5.121   0.000      -1.277259   -.5658237
    gini |    8.69435   1.989197      4.371   0.000       4.762083    12.62662
   urbmf |   2.788417   .8495378      3.282   0.001       1.109042    4.467793
 miltpop |   .0222712   .0134926      1.651   0.101      -.0044011    .0489435
```

```
------------------------------------------------------------------------------
     gaplit |   .0338037   .0190196     1.777   0.078     -.0037944    .0714018
     muslim |  -.0171654    .004497    -3.817   0.000     -.0260551   -.0082757
   lmstp90 |   .2588942    .110795     2.337   0.021      .0398735    .4779149
  Algnppci |  -1.008396   .1973407    -5.110   0.000     -1.398501   -.6182904
     Agini |   9.397534   2.202297     4.267   0.000      5.044009    13.75106
     Aurbmf |   1.980956   1.007619     1.966   0.051     -.0109163    3.972827
   Amiltpop |   .0301181   .0719627     0.419   0.676     -.1121385    .1723747
    Agaplit |   .0494775   .0367539     1.346   0.180      -.023178    .1221329
    Amuslim |  -.0216397   .0248713    -0.870   0.386     -.0708057    .0275262
   Almstp90 |  -5.257739   5.453632    -0.964   0.337     -16.03854    5.523061
      _cons |  -5.911678   2.387328    -2.476   0.014     -10.63097   -1.192382
------------------------------------------------------------------------------
```

Note that the R-squared is improved when the imputed values of the variables are permitted to have different estimated coefficients than the original variables. However, in this case the improvement is small and the F statistic actually gets smaller. Since many of the coefficients of the "A-variables" are very similar to the coefficients of the original variables in the same regression, it is not surprising that constraining the coefficients of the original `x` variable and the `Ax` variable to be identical, as is done in the `noaux` regression, only slightly worsens the fit.

After auxiliary variables are used to estimate the regression with missing values, the program tests the hypothesis that the coefficient of each zero-filled right-hand-side variable, `x`, is equal to the coefficient of its companion constructed variable, `Ax`. Failure to reject indicates that estimation with the `noaux` option would not damage the fit of the regression. This in turn would suggest that the imputation of the missing variables using the current value of n_m does not greatly alter the regression results, at least with respect to this variable. Conversely, rejection of the hypothesis that the coefficient of `x` equals the coefficient of `Ax` suggests caution in using the results for this value of n_m and this dataset. Either the imputation is bad for this variable (i.e., it is not producing accurate forecasts of the missing values), or the coefficient of `x` differs between the subset of observations for which `x` is observed and the subset for which it is not observed (i.e., the observations without missing values may have a systematically different effect on the dependent variable from those with missing values).

Next we summarize the output of the statistical tests performed by `regmsng` in response to the command:

```
regmsng logitP hirisk $RHS , nmiss(1)
```

Variable	Number of missing	H_0: Jointly zero F statistic	Prob >0	H_0: Equal one another F statistic	Prob >0
`lgnppci`	1	16.25	0.0000	1.82	0.1802
`gini`	18	9.27	0.0002	0.00	0.9614
`urbmf`	6	3.50	0.0334	0.11	0.7378
`miltpop`	2	7.73	0.0007	1.96	0.1643
`gaplit`	7	3.63	0.0296	1.75	0.1882
`muslim`	2	4.87	0.0093	1.48	0.2263

Of the eight right-hand-side variables, only `epiage` and `lmstp90` are never missing within these 132 observations. The other six variables are missing on from one to 18 observations. For each of these, the table presents two F tests calculated by `regmsng`. Since a variable like `gini` (the Gini coefficient of income inequality) is now represented by two variables in the regression, a t test on the coefficient of the original variable is not sufficient to test the hypothesis that `gini` is contributing significantly to explaining the variation in `logitP`. Therefore, `regmsng` performs a test of the hypothesis that both coefficients, that of `gini` and that of `Agini`, are jointly zero. The third column in the table shows that all six of these variables retain their statistical significance at the 5 percent level under this test.

By placing all the imputed values of `gini` in a separate variable called `Agini`, `regmsng` without the `noaux` option allows a test of the hypothesis that the coefficient of the original zero-filled variable `gini` is the same as the coefficient of the variable containing the imputed values `Agini`. If the model is stable across the observations for which `gini` was observed and those for which it is being imputed and if the imputation is relatively accurate, the coefficients of `gini` and `Agini` should not be significantly different. Our table shows that in this example none of the coefficients of the six variables with missing variables is significantly different than the coefficient of its auxiliary A-variable. This information lends confidence to the use of the extra observations. It also means that in this example the `noaux` option would not greatly worsen the fit of the regression or greatly change the values of the estimated coefficients.

Replacing the missing values with fitted values from instrumental regressions

An alternative to replacing the missing values in variable `x` by the mean of `x` is to replace them by fitted values from a regression of `x` on a set of appropriate instruments. This alternative is selected by specifying the option fitted. By analogy with

instrumental variable estimation, it is reasonable to use all available exogenous variables as instruments for predicting the value of `x` on those observations on which it is missing. Under the classical regression assumption that all right-hand-side variables are independent of the disturbance term, all the other right-hand-side variables in the regression are candidates to use as instruments to project `x` to observations on which it is missing. If other variables are available which are not among the right-hand-side variables and are independent of the disturbance term of the equation being estimated, then these additional exogenous variables can be used to improve the predictions of the missing values of included variables, by specifying those variables using the option `inst(`*varlist*`)`. To be useful in helping to estimate a missing value of the variable `x`, a variable must not be missing on the same observations on which `x` is missing. Therefore, `regmsng` only uses those right-hand-side and instrumental variables that are present on all the observations on which `x` is missing. This logic leads `regmsng` to vary the number of actual regressors from one auxiliary regressor to another, as will be obvious when the fitted option is specified on the example dataset.

If the `steps` option is specified with the `fitted` option, `regmsng` presents all the detailed results of the instrumental regressions used to impute the missing values. Otherwise, the `inst` option produces a table with one line describing the result of each of the instrumental regressions.

Here is the table that is produced when n_m is set to 3 and a vector of ten additional exogenous variables is specified in the global macro `INST`.

```
. regmsng logitP hirisk $RHS , nmiss(3) fitted inst($INST) nohist notest
                Number of   Number of
Independent      Proposed     Actual                 Mean of     Mean of
Variables  nobs Instruments Regressors   F-stat   Nonmissing   Imputed
---------  ---- ----------- ----------   ------   ----------   -------
lgnppci     116       8          14        41.82     7.7840363   1.0797078
gini        116       6          10         5.21     .44592119   .40394952
urbmf       142       6          11         7.02     1.0914893   .95682817
miltpop     145       7          12         4.47     11.26529    9.6308539
gaplit      134       9          16         9.15     15.181816   8.2030267
muslim      132       9          15         7.32     23.456944   14.433267
lmstp90     151       9          17         4.35     .3033641    .15338174
```

Note that the number of regressors varies from variable to variable. The number is determined by which variables are present on the observations that need to be imputed. The likely precision of the imputation is given by the F statistic of the instrumental variable. If this F statistic is less than four, the imputation would be poor and mean replacement should probably be used instead. The columns labeled "Number of Actual Regressors" and "nobs" give respectively the numerator and denominator degrees of freedom of the F statistic.

The second to last column of the table gives the mean of the nonmissing values of the variable. This mean would be substituted for the missing values under the mean-replacement option. The last column gives the mean of the values imputed as fitted values from the instrumental regressions. When the value in the last column differs greatly from that in the second to last, the `fitted` option is providing very different imputations than would the mean-replacement option. Note that Version 1 of `regmsng` does not allow a mix of replacement approaches on the same regression equation. It requires that all missing values be replaced either by means or, if `fitted` is specified, by fitted values.

Greene (1997) and Griliches (1986) use the term "zero-order replacement" to refer to the replacement of missing values of a variable by the mean of that variable and the term "first-order replacement" to refer to replacement by a fitted value from a regression equation. If the situation is really one of "ignorably missing" data, and the correlations among the right-hand side variables are high enough for instrumental regressions of one of them on the others to have pretty good fits, then first-order replacement should work better than zero-order replacement. But the power of first-order replacement (using `regmsng`'s `fitted` option) declines as the number of right-hand-side variables available to predict the missing values declines. Thus, when doing first-order replacement, an additional cost of increasing n_m above 1 is that higher values of n_m entail a reduction in the number of right-hand-side variables that are available for predicting and imputing the missing values.

How can one tell whether the missing values are "ignorably missing"?

Like other fundamental assumptions in the regression model, the assumption that missing values are "ignorably missing" is difficult to test. If the true values of the missing values were available, then it would be straightforward to construct a test that the estimated coefficients of the regression are different when those "missing" values are included than when they are not. If the coefficients do change significantly, then the missing data is not ignorable. Since the missing values are not available, but must be estimated from the available information, this straightforward test is impossible. However, an approximate version of the test might be constructed after replacing the missing values by consistent estimates. The power of this approximate test would improve with the accuracy of the imputation procedure, until in the limit as the imputation procedure approaches perfection, the test approaches the ideal test that could be performed if the missing values were actually available.

I have not attempted to implement such an approximate test or to explore its statistical properties. (I would appreciate hearing from any readers who have ideas on how to implement it.) However, while we wait for such a test, the analyst working with missing data can use `regmsng` in order to get a feel for whether the missing values are ignorable. In comparing the output of `regress` with that of `regmsng` for various values of `nmiss`, I would be encouraged if the estimated coefficients change very little, while their standard errors shrink.

When the estimated coefficients are sensitive to the inclusion of additional observations, there are two possible explanations. First, the missing values might be truly ignorably missing, but the inaccurate imputation procedure is substituting such wrong values for the missing values that the coefficients estimated on these poorly imputed values are distorted. On the other hand, the imputations may be accurate, but the data are not ignorably missing. Or both explanations may apply simultaneously. In this situation, ignoring the observations with missing data and reporting only the results from the complete dataset may be the easiest approach, but is not really honest. After all, the evidence available to the analyst suggests the strong possibility that the regression coefficients estimated on only the complete data are biased by the sample selection process that determined which observations would have complete data. The preferable approach is to model the sample selection process explicitly, for example by using Stata's `heckman` command. But it might also be possible to re-specify the model so that the coefficients are less sensitive to the inclusion of the extra observations. A different imputation procedure, using a different set of instrumental variables with the `fitted` option, might make the coefficients less sensitive. Or the addition of explanatory variables to the model that capture the special features of the observations being added might stabilize the coefficients.

Final thoughts on the usefulness of regmsng

The operations performed by `regmsng` are not complex. An analyst could do the entire sequence of operations with a few commands, ending by replacing the missing values by imputed values and applying `regress`. However, in view of the number of options available and the fact that missing values must be replaced, a direct approach is likely to be tedious to document and difficult to replicate. Furthermore, it is possible that the analyst would become confused about which observations are part of the original data and which have been imputed. Multiple, slightly different copies of the same dataset proliferate and litter one's hard disk. One of the most useful attributes of `regmsng` is that by default it does not alter the data. By internal use of Stata's `preserve` command, `regmsng` is guaranteed to leave the data in the condition it found them, complete with all of the original "holes." Thus, using `regmsng` the analyst can easily experiment with a variety of approaches to replacing the missing values without accumulating a large number of intermediate variables or risking confusion about the real data.

However, there are some situations in which it is desirable to retain the altered data. One such situation is when the analyst wishes to apply estimation techniques that are not part of the `regmsng` command. For example, the `aidsdata` dataset contains two observations on each of many of the countries. It is reasonable to suppose that the disturbances on pairs of observations from the same country are correlated. In order to correct the variance–covariance matrix of coefficients for this feature of the data, it is desirable to use the `cluster` option, but this option is not recognized by `regmsng`. The solution to this problem is to estimate the equation by `regmsng` with the `change` option, and then re-estimate the same equation a second time using `regress` on the changed data, but adding the `cluster` option. This procedure is used at the end of the sample `aidsdata.do` file included on the STB diskette. `aidsdata.do` also shows how to implement manually the same set of F tests of the coefficients of auxiliary variables that `regmsng` would do automatically.

Finally, it is important to note that `regmsng` makes no attempt to correct the estimated standard errors of the coefficients for the fact that some of the data has been imputed. Such a correction would typically increase the standard errors.

For a systematic exploration of the robustness of a set of regression coefficients to different values of `nmiss` and different `regmsng` options, see `aidsdata.do` and its log file, `aidsdata.log`, both of which are included on the STB diskette.

References

Greene, W. C. 1997. *Econometric Analysis.* 3d ed. Upper Saddle River, NJ: Prentice–Hall.

Griliches, Z. 1986. Economic data issues. In *Handbook of Econometrics*, vol. 3, ed. Z. Griliches and M. Intriligator. Amsterdam: North Holland.

Over, M. 1998. An exploratory analysis of the determinants of urban HIV prevalence, In *Economic Perspectives on the AIDS Epidemic: Background Papers for "Confronting AIDS"*, ed. M. Ainsworth, L. Fransen, and M. Over, Brussels: European Commission.

World Bank, 1997. *Confronting AIDS: Public Priorities in a Global Epidemic.* New York: Oxford University Press.

sg100	Two-stage linear constrained estimation

Jeroen Weesie, Utrecht University, Netherlands, J.Weesie@fss.uu.nl

Stata supports constrained estimation in a limited number of estimation commands and statistical models, namely linear regression (`cnsreg`), multinomial logistic regression (`mlogit`) and simultaneous equations (`reg3`). This insert provides an approximate method that can be used for estimation commands. It is based on the theory of equality-constrained estimation that

states, under regularity conditions, constrained optimization problems can be made first-order equivalent to a two-stage problem; see Gourieroux and Montfort (1995).

For the linear regression with linear constraints case, the equivalence is even exact. (The estimated coefficients from `cnsreg` and `regress/linest` are (should be) identical, but the estimated standard errors will differ somewhat as a consequence of the asymptotic (large sample) nature of the equivalence of constrained and two-stage estimation; the estimate of σ^2 is not adapted for the imposed constraints.) For other models the equivalence is asymptotic, i.e., valid in large samples.

In fact, the asymptotic equivalence applies even to general smooth nonlinear explicit or implicit constraints on the parameters. This can easily be employed, for instance, to estimate models with interaction effects subject to a rank condition (e.g., Goodman's $r \times c$ models, principal-components-like models, etc.). At this moment Stata does not contain an efficient tool for nonlinear GLS-estimation, and I simply can't make this large investment now.

Syntax

Let us be a little more formal now. The command `linest` has the following syntax

`linest` [`, constraints(`*clist*`) level(#) eform(`*str*`) keep modify`]

where *clist* is of the form `#[ - #][ , #[ - #] ...]`.

Description

`linest` performs linear constrained estimation after unconstrained estimation (e.g., `qreq`, `logit`, `stcox`, `heckman`, `mvreg`, ...) was performed. `linest` displays the table with the two-stage constrained estimates and their confidence intervals, and a Wald test for the hypothesis that the parameters satisfy the constraints.

`linest` can be used as a variant of `test` and `testparm` that requires the specification of the hypotheses by means of Stata's `constraints`.

The standard errors of the unconstrained estimators may be of the standard or "robust" sandwich type and the standard errors may be adjusted for clustering.

The two-stage constrained estimator is asymptotically equivalent to the one-stage constrained estimator (Gourieroux and Montfort 1995, Ch. 10). This result, however, relies on additional linearization. Thus, if a one-stage constrained estimator is available (`cnsreg`, `mlogit`, `reg3`), I expect this estimator to be better behaved in small samples.

Options

`constraints(`*clist*`)` specifies the constraint numbers of the constraints to be applied, separated by commas. Constraints are defined using the `constraint` command.

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

`eform(`*str*`)` specifies that the parameter estimates should be exponentiated. In accordance with Stata's regular behavior, the standard errors are left unchanged, and the confidence interval is exponentiated as well. The string argument of `eform` specifies the name of the column with transformed coefficients.

`keep` specifies that the results for the constrained estimates are saved for replay, and for post-estimation commands (e.g., `predict`, `test`).

`modify` specifies that the results for the unconstrained estimates are modified into the associated results for the constrained estimates. Thus, replaying the original estimation command will display the results for the constrained estimates. `modify` is not permitted for "internal" estimation commands (e.g., `logit/logistic`, `ologit`, `cox/stcox`).

Dependencies

`linest` requires the command `matginv` (Weesie 1997).

Example 1

Using the automobile data, I first want to estimate a logistic regression model for whether a car is foreign built, subject to the constraint that 6*`price+weight` adds to 0.001. (I deny you the insights into the complex combination of mechanical and international trade theory that lead me to this deep theoretical result.) In the documentation, you see that `logit` does not permit constraints. Using `logit` and `linest`, you can go ahead as follows.

```
. logit foreign price weight trunk, nolog
Logit Estimates                                   Number of obs =     74
                                                  chi2(3)       =  54.11
                                                  Prob > chi2   = 0.0000
Log Likelihood = -17.976058                       Pseudo R2     = 0.6008
------------------------------------------------------------------------------
 foreign |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
   price |   .0009307   .0003038      3.063   0.002       .0003352    .0015262
  weight |  -.0058772   .0017015     -3.454   0.001       -.009212   -.0025424
   trunk |  -.0033898   .1423407     -0.024   0.981      -.2823724    .2755927
   _cons |   9.028333   2.881154      3.134   0.002       3.381374    14.67529
------------------------------------------------------------------------------

. test 6*price + weight = 0.001
 ( 1)  6.0 price + weight = .001
           chi2(  1) =    2.74
         Prob > chi2 =    0.0979
```

`test` reports the Wald test for the constraint on `price` and `weight`, but does not report the constrained parameter estimates. After defining the linear constraint with the Stata command `constraint`, we can invoke `linest` to obtain the constrained estimates.

```
. constraint def 1  6*price + weight = 0.001
. linest, c(1)
Two-step constrained logit
                                          Dim unrestricted model =        4
                                          Dim restricted model   =        3
                                          # restrictions         =        1
                                          X2 for restrictions    =   2.7388
                                          Prob > chi2(1)         =   0.0979
 ( 1)  6.0 price + weight = .001
------------------------------------------------------------------------------
         |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
   price |    .001114   .0002829      3.938   0.000       .0005595    .0016685
  weight |  -.0056841   .0016975     -3.349   0.001       -.009011   -.0023571
   trunk |   -.104074   .1286838     -0.809   0.419      -.3562895    .1481416
   _cons |   8.477088   2.861835      2.962   0.003       2.867995    14.08618
------------------------------------------------------------------------------
```

Note that constrained estimation results are not stored and that post-estimation commands use the unconstrained model!

Also `linest` reports the dimensions of the unconstrained and constrained models (estimated via the rank of respective variance matrices, using the matrix procedures in Weesie (1997)), and also reports the Wald test for the constraint(s), and so it can also be used as a replacement for `test` if your hypotheses are represented by Stata constraints (the exception being that `linest` always presents p values based on a chi-squared distribution, and so it is less suitable for tests on coefficients for normal-based linear regression models, and for obtaining constrained estimates here; you should use `test`/`cnsreg` in this case.)

After displaying the parameter estimates table, `linest` warns the user that the results for the constrained model are not saved. Post-estimation commands such as `test`, `vce`, `predict`, and so on, are thus based on the results for the unconstrained model. See below for options that change this behavior of `linest`.

It is possible to obtain estimates for logistic regression with the coefficient of some variable fixed as in

```
. test 6*price + weight = 0.001
 ( 1)  6.0 price + weight = .001
           chi2(  1) =    2.74
         Prob > chi2 =    0.0979
. test trunk - 0.5 = 0, acc
 ( 1)  6.0 price + weight = .001
 ( 2)  trunk = .5
           chi2(  2) =   24.77
         Prob > chi2 =    0.0000
. constraint def 2 trunk - 0.5 = 0
```

```
. linest, c(1,2)
Two-step constrained logit
                                          Dim unrestricted model =        4
                                          Dim restricted model   =        2
                                          # restrictions         =        2
                                          X2 for restrictions    =  24.7748
                                          Prob > chi2(2)         =   0.0000
 ( 1)  6.0 price + weight = .001
 ( 2)  trunk = .5
------------------------------------------------------------------------------
         |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
   price |   .0011159   .0002829      3.944   0.000       .0005614    .0016704
  weight |  -.0056955   .0016975     -3.355   0.001      -.0090225   -.0023686
   trunk |          .5          .         .       .              .           .
   _cons |   1.631003   2.462352      0.662   0.508      -3.195117    6.457124
------------------------------------------------------------------------------
```

Example 2

`linest` also supports linear constraints for multi-equation models such as `heckman`, `gnbreg`, etc. `linest` only uses the unconstrained parameter estimate and an associated variance estimate; it does not require any knowledge of the model that was being estimated. In multi-equation models, constraints may be imposed on parameters "within an equation," but also "between" equations. We illustrate this with Monte Carlo data that are generated from a Heckman model in which the response is predicted by `x1`, `x2`, and `x3`, and participation is modeled with `z1`, `x1`, and `x2`.

```
. set obs 2000
. for x1 x2 x3 z1 z2 e1 e2 , lt(any): gen @ = invnorm(uniform())
  (output omitted)
. gen y = cond(z1+x1+x2+e1+e2>0, x1+x2+x3+e1, .)
(977 missing values generated)
```

Note that the random terms in the equations are correlated with correlation $1/\sqrt{2} = 0.71$. To compute normal-based maximum-likelihood estimates for the Heckman model, we have to set up response and participation equations, and invoke `heckman`.

```
. eq y x1 x2 x3
. eq p: z1 x1 x2
. heckman y p
Heckman selection model                           Number of obs    =    2000
                                                  Model chi2(7)    = 1480.61
                                                  Prob > chi2      =  0.0000
Log Likelihood =  -2010.9599662
------------------------------------------------------------------------------
       y |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
y        |
      x1 |   .9424194   .0361888     26.042   0.000       .8714907    1.013348
      x2 |    1.02492   .0379274     27.023   0.000       .9505837    1.099256
      x3 |   1.002327   .0284472     35.235   0.000       .9465714    1.058082
   _cons |   .0093015   .0511742      0.182   0.856      -.0909982    .1096011
---------+--------------------------------------------------------------------
p        |
      z1 |   1.002391   .0486463     20.606   0.000       .9070462    1.097736
      x1 |   .9538658   .0492889     19.353   0.000       .8572614     1.05047
      x2 |    1.04856   .0512989     20.440   0.000       .9480164    1.149105
   _cons |  -.0051028   .0380191     -0.134   0.893      -.0796189    .0694132
---------+--------------------------------------------------------------------
_athrho  |
   _cons |   .8740199   .0907615      9.630   0.000       .6961307    1.051909
---------+--------------------------------------------------------------------
_lnsigma |
   _cons |   .0101341   .0257162      0.394   0.694      -.0402687    .0605369
------------------------------------------------------------------------------
     rho      0.70341               [_athrho]_cons  = atanh(rho)
   sigma    1.0101856               [_lnsigma]_cons = ln(sigma)
  lambda    .71057545    .057762
```

We now want to impose a *within-equation constraint* (the response effects of `x2` and `x3` are equal) and a *between-equation constraint* (the response effect of `x1` equals the participation effect of `x1`).

```
. constraint def 1 [y]x2 = [y]x3
. constraint def 2 [y]x1 = [p]x1
. linest, c(1 2)
Two-step constrained heckman
                                          Dim unrestricted model =       10
                                          Dim restricted model   =        8
                                          # restrictions         =        2
                                          X2 for restrictions    =   0.3080
                                          Prob > chi2(2)         =   0.8573
 ( 1)  [y]x2 - [y]x3 = 0.0
 ( 2)  [y]x1 - [p]x1 = 0.0
------------------------------------------------------------------------------
             |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
-------------+----------------------------------------------------------------
y            |
          x1 |   .9436947   .0309322     30.508   0.000       .8830686    1.004321
          x2 |   1.011542   .0223753     45.208   0.000       .9676871    1.055397
          x3 |   1.011542   .0223753     45.208   0.000       .9676871    1.055397
       _cons |   .0144706   .0428663      0.338   0.736      -.0695458     .098487
-------------+----------------------------------------------------------------
p            |
          z1 |   .9983908   .0421885     23.665   0.000       .9157029    1.081079
          x1 |   .9436947   .0309322     30.508   0.000       .8830686    1.004321
          x2 |   1.039522   .0463578     22.424   0.000        .948662    1.130381
       _cons |  -.0042183   .0379049     -0.111   0.911      -.0785106    .0700741
-------------+----------------------------------------------------------------
_athrho      |
       _cons |    .870172   .0828165     10.507   0.000       .7078546    1.032489
-------------+----------------------------------------------------------------
_lnsigma     |
       _cons |   .0095379   .0242137      0.394   0.694      -.0379201    .0569959
------------------------------------------------------------------------------
```

Again, constrained estimation results are not stored and post-estimation commands use the unconstrained model!

`linest` reports that the constraint coefficients of `x2` and `x3` both equal 1.01, while the response and participation effects of `x1` both equal 0.94. The Wald test for these two constraints is 0.3080, and so the equality constraints cannot be rejected for any reasonable significance level.

It is also possible to impose the linear constraints sequentially (compare the `accumulate` option with `test`). Specifying the option `keep` means that `linest` is now interpreted as a proper estimation command, so that the results of `linest` "replace" the estimation results by `heckman`.

(*Output on next page*)

```
. linest, c(1) keep
Two-step constrained heckman
                                          Dim unrestricted model =       10
                                          Dim restricted model   =        9
                                          # restrictions         =        1
                                          X2 for restrictions    =   0.2255
                                          Prob > chi2(1)         =   0.6349
 ( 1)  [y]x2 - [y]x3 = 0.0
------------------------------------------------------------------------------
         |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
y        |
      x1 |   .9387718   .0353642     26.546   0.000       .8694592    1.008084
      x2 |   1.010483   .0226769     44.560   0.000       .9660372    1.054929
      x3 |   1.010483   .0226769     44.560   0.000       .9660372    1.054929
   _cons |   .0195626   .0463885      0.422   0.673      -.0713571    .1104824
---------+--------------------------------------------------------------------
p        |
      z1 |   1.005144   .0482996     20.811   0.000       .9104788     1.09981
      x1 |   .9547038   .0492573     19.382   0.000       .8581614    1.051246
      x2 |   1.045564   .0509095     20.538   0.000       .9457837    1.145345
   _cons |  -.0038469    .037927     -0.101   0.919      -.0781824    .0704886
---------+--------------------------------------------------------------------
_athrho  |
   _cons |    .862237   .0873039      9.876   0.000       .6911244    1.033349
---------+--------------------------------------------------------------------
_lnsigma |
   _cons |   .0075855   .0251499      0.302   0.763      -.0417075    .0568785
------------------------------------------------------------------------------
```

`linest` saves results as an estimation command. Beware (see online help!) that `chi2`/`ll`/`pr2` and so on may been modified and that `predict` now returns linear predictors (`index`).

The Wald test for this one-dimensional constraint is 0.2255. We can now impose a second one-dimensional constraint to the constrained estimator.

```
. linest, c(2)
Two-step constrained linest
                                          Dim unrestricted model =        9
                                          Dim restricted model   =        8
                                          # restrictions         =        1
                                          X2 for restrictions    =   0.0825
                                          Prob > chi2(1)         =   0.7740
 ( 1)  [y]x1 - [p]x1 = 0.0
------------------------------------------------------------------------------
         |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
y        |
      x1 |   .9436947   .0309322     30.508   0.000       .8830686    1.004321
      x2 |   1.011542   .0223753     45.208   0.000       .9676871    1.055397
      x3 |   1.011542   .0223753     45.208   0.000       .9676871    1.055397
   _cons |   .0144706   .0428663      0.338   0.736      -.0695458     .098487
---------+--------------------------------------------------------------------
p        |
      z1 |   .9983908   .0421885     23.665   0.000       .9157029    1.081079
      x1 |   .9436947   .0309322     30.508   0.000       .8830686    1.004321
      x2 |   1.039522   .0463578     22.424   0.000        .948662    1.130381
   _cons |  -.0042183   .0379049     -0.111   0.911      -.0785106    .0700741
---------+--------------------------------------------------------------------
_athrho  |
   _cons |    .870172   .0828165     10.507   0.000       .7078546    1.032489
---------+--------------------------------------------------------------------
_lnsigma |
   _cons |   .0095379   .0242137      0.394   0.694      -.0379201    .0569959
------------------------------------------------------------------------------
```

Thus, in effect, we have obtained a three-stage estimator for Heckman's model with two linear constraints. The effect of imposing constraints 1 and 2 sequentially yields the same parameters estimates and standard errors as imposing these two constraints simultaneously. According to the theory of two-stage estimation, this holds true quite generally if the constraints are linear. (For nonlinear constraints, the equality only holds asymptotically.)

In some applications, it may be useful to translate the results from the constrained estimation by `linest` back to the original estimation command—`heckman` in this case. This is accomplished with the option `modify`.

```
(after heckman)
. linest, c(1 2) modify
  (output omitted)
```

The output of `linest` is omitted, as it is almost identical to the invocation of `linest` without the `modify` option. Only the warning message after displaying the constrained parameter estimates is slightly different. We can now let `heckman` replay its output. Note that the title now includes the qualification "(constrained)".

```
. heckman
Heckman selection model (constrained)              Number of obs    =    2000
                                                   Model chi2(5)    = 1480.30
                                                   Prob > chi2      =  0.0000
Log Likelihood =  -2011.2679524
------------------------------------------------------------------------------
             |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
-------------+----------------------------------------------------------------
y            |
          x1 |   .9436947   .0309322     30.508   0.000       .8830686    1.004321
          x2 |   1.011542   .0223753     45.208   0.000       .9676871    1.055397
          x3 |   1.011542   .0223753     45.208   0.000       .9676871    1.055397
       _cons |   .0144706   .0428663      0.338   0.736      -.0695458     .098487
-------------+----------------------------------------------------------------
p            |
          z1 |   .9983908   .0421885     23.665   0.000       .9157029    1.081079
          x1 |   .9436947   .0309322     30.508   0.000       .8830686    1.004321
          x2 |   1.039522   .0463578     22.424   0.000        .948662    1.130381
       _cons |  -.0042183   .0379049     -0.111   0.911      -.0785106    .0700741
-------------+----------------------------------------------------------------
_athrho      |
       _cons |    .870172   .0828165     10.507   0.000       .7078546    1.032489
-------------+----------------------------------------------------------------
_lnsigma     |
       _cons |   .0095379   .0242137      0.394   0.694      -.0379201    .0569959
------------------------------------------------------------------------------
         rho      0.70146               [_athrho]_cons  = atanh(rho)
       sigma    1.0095835               [_lnsigma]_cons = ln(sigma)
      lambda    .70818397   .0521798
```

Comparing the output from the `heckman` command before constraints were imposed, and after the constraints were imposed via `linest, c( ) modify`, we see that the `heckman` command now displays the constrained estimates and associated standard errors. The log-likelihood value has increased from −2010.95 to −2011.26 and the `chi2` test statistic (i.e., the likelihood-ratio test statistic for the estimated model against the constant-only model) has decreased from 1480.61 with 7 degrees of freedom to 1480.30 with 5 degrees of freedom. Where do these changes come from? `linest` has tried to approximate the change in the log-likelihood value and the `chi2` statistic that result from imposing the constraints using the value from the Wald test statistic for these constraints. This approximation is based on the asymptotic equivalence of the Wald test and the likelihood-ratio test *if the constrained model is true*. Thus, we would expect that the log-likelihood value for the constrained model would be larger by roughly half the value of the Wald statistic. The `chi2` test is approximated in a similar way. (Where appropriate, `linest` also modifies the pseudo-R2.) The change in the number of degrees of freedom is simply the number of (independent) restrictions.

Note: For reasons that involve the implementation of internal estimation commands such as `regress` and `logit` (and hence also `logistic` that is just a shell around `logit`), `linest` only supports the `modify` option with external estimation commands.

Acknowledgment

This research was supported by grant PGS 370-50 by the Netherlands Organization for Scientific Research. Vince Wiggens (StataCorp) was helpful with suggestions on linking `linest` with the Stata estimation engine.

References

Gourieroux, C. and A. Monfort. 1995/1989. *Statistics and Econometric Models*. 2 Volumes. Cambridge: Cambridge University Press.

Magnus, J. R. and H. Neudecker. 1988. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. New York: John Wiley & Sons.

Silvey, S. D. 1970. *Statistical Inference*. London: Chapman and Hall.

Weesie, J. 1997. dm49: Some new matrix commands. *Stata Technical Bulletin* 39: 17–20. Reprinted in *Stata Technical Bulletin Reprints*, vol. 7, pp. 43–48.

sg101	Pairwise comparisons of means, including the Tukey wsd method

John R. Gleason, Syracuse University, loesljrg@ican.net

This insert presents two commands, `prcomp` and `prcompw`, for pairwise comparisons of means. `prcomp` is a command line program, while `prcompw` uses Stata's dialog programming features available only on the Windows and Macintosh platforms. These commands are otherwise equivalent; each permits several methods of pairwise comparisons, including Tukey's *wsd* procedure, and each provides both tabular and graphical display of results.

First, a quick review of relevant background. Consider a set of r sample means $\overline{Y}_1, \overline{Y}_2, \ldots, \overline{Y}_r$ based on samples of sizes $n_1, n_2, \ldots, n_r$. Either exactly, or approximately via the central limit theorem, $\overline{Y}_j \sim N(\mu_j, \sigma_j^2/n_j)$ for any $j \in R = \{1, 2, \ldots, r\}$, where μ_j and σ_j^2 are the mean and variance of the variable Y in the jth sample. Then, for any pair of distinct subscripts (denoted $j, j' \in R$), $\overline{Y}_j - \overline{Y}_{j'}$ has mean $E(\overline{Y}_j - \overline{Y}_{j'}) = \mu_j - \mu_{j'}$, and variance

$$\sigma^2_{j,j'} = \sigma_j^2/n_j + \sigma_{j'}^2/n_{j'} \tag{1}$$

provided that $\overline{Y}_j$ and $\overline{Y}_{j'}$ are uncorrelated. In the common case of a simple analysis of variance layout, it is assumed that the $\overline{Y}_j$ are independent and that $\sigma_1 = \sigma_2 = \cdots = \sigma_r = \sigma$, so that (1) becomes

$$\sigma^2_{j,j'} = \sigma^2(1/n_j + 1/n_{j'}) \tag{2}$$

In some special cases when the $\overline{Y}_j$ are dependent, an expression resembling (2) still holds for all $j, j' \in R$,

$$\sigma^2_{j,j'} = \sigma_0^2(1/n_j + 1/n_{j'}) \tag{3}$$

where σ_0^2 is a certain variance. This can occur when the $\overline{Y}_j$ satisfy a kind of sphericity condition, the simplest version of which is the compound symmetry condition of the classic repeated measures ANOVA model.

In any case, a pairwise comparisons method amounts to computing

$$\overline{Y}_j - \overline{Y}_{j'} \quad \text{and} \quad A_{j,j'} = d_{j,j'}\widehat{\sigma}_{j,j'} \quad \text{for one or more pairs } j, j' \in R \tag{4}$$

where $\widehat{\sigma}_{j,j'}$ is an estimate of $\sigma_{j,j'}$, the standard error of $\overline{Y}_j - \overline{Y}_{j'}$, and $d_{j,j'}$ is a constant that sets the confidence or significance level. Confidence intervals for $\mu_j - \mu_{j'}$ are given by $(\overline{Y}_j - \overline{Y}_{j'}) \pm A_{j,j'}$; a test of H_0: $\mu_j = \mu_{j'}$ rejects H_0 if $|\overline{Y}_j - \overline{Y}_{j'}| > A_{j,j'}$. Thus, the term $A_{j,j'}$ is the confidence interval half-width, or the "critical difference," depending on context. Tukey's *wsd* method is the classic special case: $\sigma^2_{j,j'}$ is given by (2) or (3) so that $\widehat{\sigma}_{j,j'} = \widehat{\sigma}\sqrt{1/n_j + 1/n_{j'}}$ or $\widehat{\sigma}_{j,j'} = \widehat{\sigma}_0\sqrt{1/n_j + 1/n_{j'}}$; for all $j, j' \in R$, $\sqrt{2}d_{j,j'} = q_p(r, \nu)$, the pth quantile of the studentized range distribution with parameters r and ν; ν is the degrees of freedom for the estimator $\widehat{\sigma}$ or $\widehat{\sigma}_0$. It is assumed that $\widehat{\sigma}$ or $\widehat{\sigma}_0$ is independent of the $\overline{Y}_j - \overline{Y}_{j'}$ and that $\nu\widehat{\sigma}^2/\sigma^2 \sim \chi^2(\nu)$ or that $\nu\widehat{\sigma}_0^2/\sigma_0^2 \sim \chi^2(\nu)$. The entire set of $r(r-1)/2$ confidence intervals has simultaneous confidence level at least p; the corresponding tests have simultaneous significance level at most $1 - p$. If the n_j are unequal, this is often called the Tukey–Kramer method.

Other variations are possible, most of which set $d_{j,j'} = t_P(\nu)$, the Pth quantile of the student t distribution, for some P. In the Tukey *wsd* setting, choosing $d_{j,j'} = t_{(1+p)/2}(\nu)$ gives intervals with individual confidence p and tests with individual significance $1 - p$. The value p can then be adjusted upward to control the simultaneous level, say using the Bonferroni Inequality. In the case of uncorrelated means with unequal variance, one might use (1) to obtain $\widehat{\sigma}^2_{j,j'} = s_j^2/n_j + s_{j'}^2/n_{j'}$ where s_j^2 and $s_{j'}^2$ are the relevant sample variances. This is the Welch estimate of $\sigma^2_{j,j'}$ which has degrees of freedom $\widehat{\nu}_{j,j'}$ given by Satterthwaite's approximation. (The `ttest` command with the `unequal` option uses the Welch–Satterthwaite approach.) One could then take $d_{j,j'} = t_{(1+p)/2}(\widehat{\nu}_{j,j'})$ and adjust p as desired.

`prcomp` and `prcompw` allow all these variations. Their strategy is to compute (4) for all $j, j' \in R$ and save the $\overline{Y}_j - \overline{Y}_{j'}$ and $A_{j,j'}$ in a matrix. Confidence intervals and tests can then be explored at will. The Tukey *wsd* method requires the command `qsturng` (Gleason 1998) for computing $q_p(r, \nu)$; using $t_P(\nu)$ to set the value of $d_{j,j'}$ requires only Stata's internal function `invt()`.

Paired comparisons from the command line

The syntax of the `prcomp` command is

```
prcomp  yvar xvar [weight] [if exp] [in range] [ , anova graph level(#) nolabel nolegend
        nolist nomeans nu(#) order(ord) refresh sigma( {σ̂ | σ̂0} ) stdrng
        tukey test unequal graph_options ]
```

where *yvar* is the variable whose means are the $\overline{Y}_j$ and *xvar* is the variable (say X) that defines the r groups or samples. The data need not be sorted by X, and the current sort order is not disturbed. `aweights` and `fweights` are allowed. By default, `prcomp` calculates the elements of (4) if necessary, lists the $\overline{Y}_j$, n_j, and $s_j/\sqrt{n_j}$, and then lists the $r(r-1)/2$ confidence intervals $(\overline{Y}_j - \overline{Y}_{j'}) \pm A_{j,j'}$. Also by default, $\widehat{\sigma}_{j,j'} = \widehat{\sigma}\sqrt{1/n_j + 1/n_{j'}}$ where $\widehat{\sigma} = \sqrt{MS_e}$ from a one-way ANOVA of Y classified by X, and the individual confidence level is set by the macro `S_level`. (Often, `S_level` is `95`, and then `prcomp` sets $d_{j,j'} = t_{.975}(\nu)$.)

Options

`anova` displays the one-way ANOVA summary from which $\sqrt{MS_e}$ is taken, even if $\sigma_{jj'}$ is estimated in a different way.

`graph` specifies that tables are also to be rendered in graphic form.

`level(#)` controls the confidence (or, significance) level; # may be supplied as a proportion (`0.98333`), or as a percentage (`98.333`) in which case it is divided by `100`. For $0 < p < 1$, `level(`p`)` chooses $d_{j,j'} = t_{(1+p)/2}(\nu)$ by default, and $d_{j,j'} = q_p(r,\nu)/\sqrt{2}$ if the `tukey` or `stdrng` option is present.

`nolabel` suppresses a value label and forces levels of the variable X to be labeled by their numeric value.

`nolegend` will suppress the explanatory message seen near the top of any graph produced by the command.

`nolist` suppresses the display of confidence intervals (or tests, if `test` is present).

`nomeans` suppresses the display of the $\overline{Y}_j$ and $s_j/\sqrt{n_j}$.

`nu(#)` specifies a value for degrees of freedom.

`order(`*ord*`)` allows the display of means and confidence intervals or tests in several orders; its argument is one of the words `label`, `mean`, or `natural`. Only the first character is significant; uppercase yields ascending and lowercase gives descending order. Thus, the default ordering is equivalent to `order(N)`. The option `order(L)` presents results with the levels of X sorted so that their labels are in ascending order; `order(M)` sorts that the $\overline{Y}_j$ are in ascending order.

`refresh` forces recalculation of all relevant quantities, exactly as on the initial call to `prcomp` during the current session.

`sigma()` allows the user to supply an estimate of σ_0 from a model known (or suspected) to be more appropriate than the simple one-way ANOVA.

`stdrng` and `tukey` are identical and require a homogeneity of variance assumption: $\sigma_{j,j'}$ must be estimated from (2) or (3), not (1).

`test` switches from confidence intervals to significance tests, displaying the $\overline{Y}_j - \overline{Y}_{j'}$ and $A_{j,j'}$ in triangular matrix format, and marking "significant" differences with '*'.

`unequal` specifies that the equal variance assumption should not be made in the analysis.

graph_options includes most `graph twoway` options; these are ignored when the `graph` option is absent. Any `graph` options suitable for `twoway` style can be supplied, except for `b2title`, `xscale`, `ylabel`, and `yreverse`.

Example 1

Consider a small dataset that appears in Table 12.10.2 of Snedecor and Cochran (1980, 229):

```
. use perpupil, replace
(1977 Public School Expenditures)

. describe
Contains data from perpupil.dta
  obs:            48                          1977 Public School Expenditures
 vars:             2                          23 Aug 1998 09:21
 size:           432 (99.9% of memory free)   * _dta has notes
-------------------------------------------------------------------------------
   1. expend      float  %9.0g                 1977 Per Pupil Expenditures ($)
   2. region      byte   %8.0g       region    Geographic region of the U.S.
-------------------------------------------------------------------------------
Sorted by:
```

The dataset `perpupil.dta` is included with this insert. The observations are the lower 48 of the United States. The variable `expend` contains the per-pupil expenditures (in $1000) by these states for public school education during 1977; `region` classifies the 48 states into five geographic regions.

```
. prcomp expend region
                Pairwise Comparisons of Means
Response variable (Y): expend      1977 Per Pupil Expenditures ($)
Group variable (X):    region      Geographic region of the U.S.
  Group variable (X): region        Response variable (Y): expend
------------------------------    ------------------------------
      Level         Label              n         Mean         S.E.
----------------------------------------------------------------
          1        N.East             10        1.763     .1113558
          2        S.East              7         1.33     .0691788
          3        S.Cent              9     1.178889     .0250801
          4        N.Cent             11     1.562727     .0638192
          5        Mt.Pac             11     1.507273     .0605894
----------------------------------------------------------------
Individual confidence level: 95%    (t method)
Homogeneous error SD = .2269174, degrees of freedom = 43
                                                                      95%
Level(X)    Mean(Y)   Level(X)    Mean(Y)      Diff Mean     Confidence Limits
--------------------------------------------------------------------------------
  S.East       1.33     N.East      1.763          -.433   -.6585188  -.2074811
  S.Cent   1.178889     N.East      1.763      -.5841111   -.7943742   -.373848
                        S.East       1.33      -.1511111   -.3817312   .0795089
  N.Cent   1.562727     N.East      1.763      -.2002727   -.4002224   -.000323
                        S.East       1.33       .2327272    .0114696   .4539849
                        S.Cent   1.178889       .3838384    .1781523   .5895244
  Mt.Pac   1.507273     N.East      1.763      -.2557273   -.4556769  -.0557776
                        S.East       1.33       .1772727    -.043985   .3985304
                        S.Cent   1.178889       .3283839    .1226978   .5340699
                        N.Cent   1.562727      -.0554545   -.2505854   .1396764
--------------------------------------------------------------------------------
```

The display above illustrates `prcomp`'s default behavior. The tables of means and confidence intervals are presented in "natural" order, so that values of `region` appear in ascending numeric order.

`prcomp` will also graph its confidence intervals. To list and graph 95% simultaneous confidence intervals using the Tukey *wsd* method, with the regional means of `expend` arranged in descending order, use a command such as

```
. prcomp expend region, tukey ord(m) nomean
>        gr xlab(-1,-.75,-.5,-.25,0,.25) xline(0)
                Pairwise Comparisons of Means
Response variable (Y): expend      1977 Per Pupil Expenditures ($)
Group variable (X):    region      Geographic region of the U.S.
Simultaneous confidence level: 95%    (Tukey wsd method)
Homogeneous error SD = .2269174, degrees of freedom = 43
                                                                      95%
Level(X)    Mean(Y)   Level(X)    Mean(Y)      Diff Mean     Confidence Limits
--------------------------------------------------------------------------------
  N.Cent   1.562727     N.East      1.763      -.2002727   -.4825412   .0819958
  Mt.Pac   1.507273     N.East      1.763      -.2557273   -.5379958   .0265412
                        N.Cent   1.562727      -.0554545   -.3309204   .2200113
  S.East       1.33     N.East      1.763          -.433   -.7513644  -.1146355
                        N.Cent   1.562727      -.2327272   -.5450761   .0796217
                        Mt.Pac   1.507273      -.1772727   -.4896216   .1350762
  S.Cent   1.178889     N.East      1.763      -.5841111    -.880939  -.2872833
                        N.Cent   1.562727      -.3838384   -.6742049  -.0934719
                        Mt.Pac   1.507273      -.3283839   -.6187504  -.0380174
                        S.East       1.33      -.1511111    -.476677   .1744547
--------------------------------------------------------------------------------
```

(*Graph on next page*)

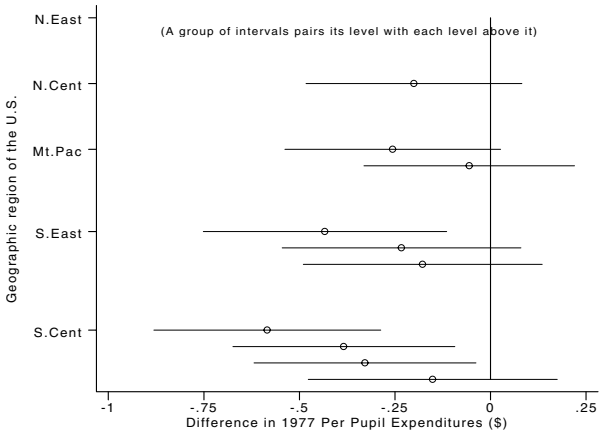


Figure 1. Graphing confidence intervals.

The confidence intervals are plotted as groups of horizontal error bars with the $\overline{Y}_j - \overline{Y}_{j'}$ as the plot symbols centered on the bars. Each group of intervals compares its own level of X with all levels of X shown above it on the vertical axis; the ordering within groups is the same as that on the vertical axis, read from the top downward. Thus the plot is ordered to match the printed list of confidence intervals.

With the `test` option, `prcomp` switches to presenting significance tests. By way of illustration, note from the display of the $\overline{Y}_j$ and their standard errors (near the outset of this example) that homogeneity of variance across regions is an untenable assumption. It might be prudent to estimate the standard errors of the $\overline{Y}_j - \overline{Y}_{j'}$ from (1) using the Welch approach, and use Satterthwaite's estimate of the associated degrees of freedom. Choosing individual significance level $0.05/10 = 0.005$ then gives paired comparisons of the regional means more suited to the apparent heterogeneity of variance, and which have simultaneous significance level no greater than 0.05:

```
. prcomp expend region, test uneq level(99.5) nomean order(M) gr xlab(1,2,3,4)
                  Pairwise Comparisons of Means
Response variable (Y): expend      1977 Per Pupil Expenditures ($)
Group variable (X):    region      Geographic region of the U.S.
Individual significance level: .5%    (t method)
Welch standard errors, Satterthwaite approximate degrees of freedom
          (Row Mean - Column Mean) / (Critical Diff)
Mean(Y) |   1.1789      1.33    1.5073    1.5627
Level(X)|   S.Cent    S.East    Mt.Pac    N.Cent
--------+------------------------------------------
    1.33|   .15111
  S.East|   .28746
        |
  1.5073|   .32838*   .17727
  Mt.Pac|   .22038    .30645
        |
  1.5627|   .38384*   .23273    .05545
  N.Cent|   .23146    .31171    .27759
        |
   1.763|   .58411*     .433    .25573    .20027
  N.East|   .40982    .43526    .42154    .42431
        |
```

(*Graph on next page*)

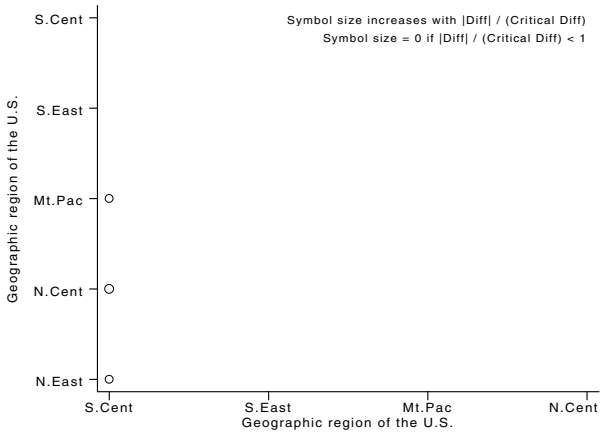


Figure 2. Graphical representation of significance tests.

In the table above, the upper value in each entry is a difference $\overline{Y}_j - \overline{Y}_{j'}$, the lower value is the corresponding critical difference $A_{j,j'}$. Significant differences are marked by an asterisk (*), color highlighted if possible. The `graph` option renders the table in graphic form: significant differences (only) are plotted with a symbol whose size reflects the magnitude of the ratio $|\overline{Y}_j - \overline{Y}_{j'}|/A_{j,j'}$. (The symbols in Figure 2 are of equal size because the three ratios are roughly equal in size.) The rows and columns of the graph match those of the table, both are controlled by the `order()` option. The `nolegend` option will again suppress the explanatory message seen at the top of Figure 2.

Example 2

`tailflip.dta` is taken from Table 12.8 of Bliss (1967, 362). The data result from a one-way repeated measures design in which each of 10 rats was tested on each of nine nonconsecutive April days.

```
. use tailflip
(Tail-flip reaction time in rats)
. describe
Contains data from tailflip.dta
  obs:           90                          Tail-flip reaction time in rats
 vars:            3                          22 Aug 1998 21:26
 size:          900 (95.8% of memory free)    _dta has notes
-------------------------------------------------------------------------------
   1. rtime       float  %8.0g                 Reaction time (log sec. - 0.6)
   2. rat         byte   %8.0g                 Rat no.
   3. day         byte   %8.0g                 Day (in April)
-------------------------------------------------------------------------------
Sorted by:
```

The variable `rtime` is the total time (over two trials) for the rat to move his tail out of a strong light beam. The times have been transformed as log(# seconds)$-$0.6. The classic one-way repeated measures ANOVA is then

```
. anova rtime rat day
                           Number of obs =      90     R-squared     =  0.5341
                           Root MSE      = .117186     Adj R-squared =  0.4241
                  Source |  Partial SS    df       MS           F     Prob > F
              -----------+----------------------------------------------------
                   Model |  1.13361046    17  .066682968       4.86     0.0000
                         |
                     rat |  .341506676     9  .037945186       2.76     0.0078
                     day |  .792103786     8  .099012973       7.21     0.0000
                         |
                Residual |  .988739514    72  .013732493
              -----------+----------------------------------------------------
                   Total |  2.12234998    89  .023846629
```

If the underlying ANOVA model can be trusted, the RMSE ($\widehat{\sigma}_0 = .117186$) and error degrees of freedom ($\nu = 72$) can be used for making paired comparisons of the 9 daily means. Commands such as

```
. global Sigma = _result(9)
. global DF = _result(5)
```

conveniently capture these values for this purpose, as we illustrate using the command `prcompw`.

A dialog form of prcomp (Windows and Macintosh only)

The command `prcompw` creates a dialog box with all of the abilities of `prcomp`; its syntax is just

`prcompw` [*varlist*]

since all of the `prcomp` options are provided as dialog controls. Launching `prcompw` with `tailflip.dta` in memory produces a dialog box resembling Figure 3.

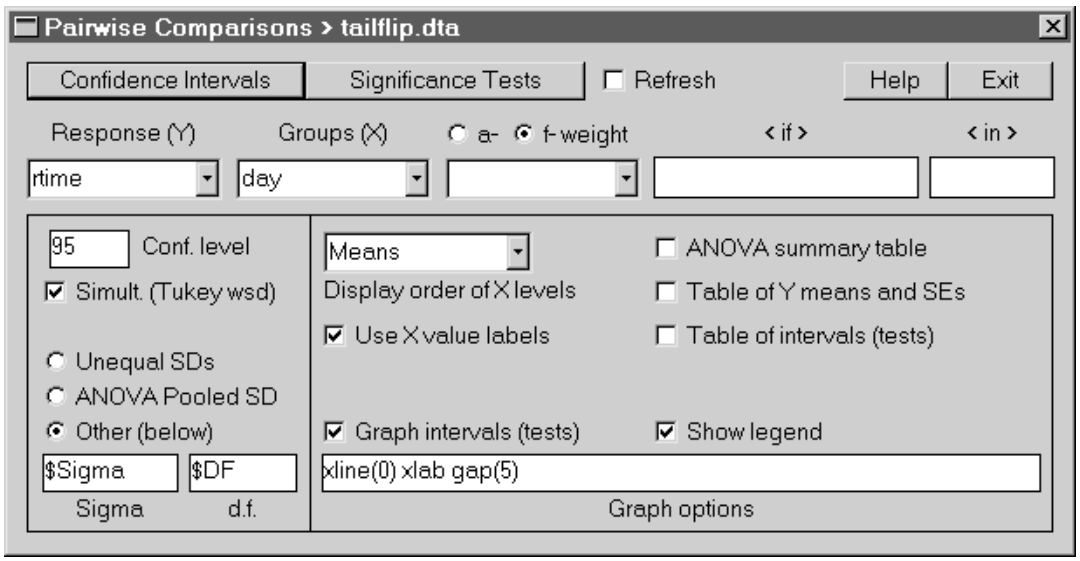


Figure 3. An example dialog box.

The purpose of most of the controls in the dialog should by now be obvious. Clicking the "Confidence Intervals" button in Figure 3 would have the same effect as issuing

```
. prcomp rtime day, sigma($Sigma) nu($DF) tukey ord(M) nolist
>        nomeans gr xline(0) xlab gap(5)
```

from the command line, thus printing

```
                  Pairwise Comparisons of Means
Response variable (Y): rtime       Reaction time (log sec. - 0.6)
Group variable (X):    day         Day (in April)
Simultaneous confidence level: 95%    (Tukey wsd method)
Homogeneous error SD = .1171857, degrees of freedom = 72
```

on the *Results* screen, and then displaying the graph shown in Figure 4. Plainly, there is something unusual about the data for April 20; Bliss (1967) does not comment on this feature.

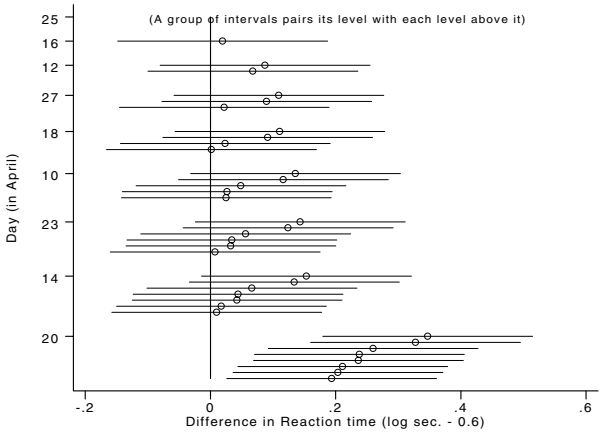


Figure 4. Graphical representation of confidence intervals.

Remarks

1. It is sometimes convenient to present the levels of X in an arbitrary order, perhaps corresponding to some anticipated set of results. One way to achieve this is to create a new X variable whose natural order traverses the desired sequence. Another is to define a value label whose labels have the right sort order, assign it (temporarily) to X and then supply the option `order(L)` or `order(l)`.

2. In the graph produced by the `test` option (see Figure 2), symbols will be plotted at $1, 2, \ldots, r-1$ on the abscissa, regardless of the actual numeric values of the variable X. Thus, `graph` options such as `xlabel`, `xline`, etc. must refer to `1,2,`..., not to the underlying values of X.

3. The left-quote (`) and double-quote (``) characters must be avoided in supplying the *graph_options*. This is a limitation of Stata's macros.

4. The options `sigma()` and `nu()` exist primarily for the reason shown in Example 2: To supply an estimate of σ_0 from a model known (or suspected) to be more appropriate than the simple one-way ANOVA. However, these options can also be used to compare cell means along one factor of a a multi-factor design, while estimating σ from all cells in the design. It is, as elsewhere, the user's responsibility to ensure that the values supplied in the `sigma()` and `nu()` options are actually sensible. In particular, the use of the classic repeated measures ANOVA in Example 2 should not be construed as advocacy for this model.

5. `prcomp` and `prcompw` do most of their work by calling on accessory programs (whose names begin with `prcmp`) which must also be installed. The intent is to encourage exploration of pairwise mean differences by providing a variety of options for defining the $A_{j,j'}$, and for presentation of results. To this end, basic computations are saved in two matrices and a set of global macros. These objects are left in memory and updated only when necessary, say when the choice of Y, X, or definition for $A_{j,j'}$ is changed. Ordinarily this logic is effective but it can be fooled, say if one of the global macros becomes corrupted, or if the data are edited after using `prcomp` or `prcompw`. The `refresh` option (checkbox, in `prcompw`) exists for this purpose: it forces recalculation of all relevant quantities, exactly as on the initial call to `prcomp` during the current session.

6. For the intensely curious: The triplets $(n_j - 1, \overline{Y}_j, s_j/\sqrt{n_j})$ are saved in the first r rows of an $(r+1)\times 3$ matrix `PrCmp0`; the last row holds the one-way ANOVA estimate of σ, along with its degrees of freedom. The $r(r-1)/2$ pairs $(\overline{Y}_j - \overline{Y}_{j'}, A_{j,j'})$ are stored in the $r \times r$ matrix `PrCmp1` with the $A_{j,j'}$ above the diagonal. The macro `PrCmp0` contains a command line for `prcomp`, less any options—information that defines the matrix `PrCmp0`. The macro `PrCmp1` contains the value of the `level()` option, the distribution used for setting $d_{j,j'}$ and, if the `unequal` option is absent, the estimate $\widehat{\sigma}$ (or $\widehat{\sigma}_0$) and its degrees of freedom; this information determines the matrix `PrCmp1`. The macro `PrCmp0K` contains the numeric values of the levels of X; these are also row names in the matrices `PrCmp0` and `PrCmp1`. The macro `PrCmp0X` holds two copies of $R = \{1, 2, \ldots, r\}$; the first copy indexes the levels of X with value labels in ascending order, the second copy indexes X with the $\overline{Y}_j$ in ascending order. Each accessory program has a terse comment explaining its purpose and arguments.

Acknowledgment

This project was supported by a grant R01-MH54929 from the National Institute on Mental Health to Michael P. Carey.

References

Bliss, C. I. 1967. *Statistics in Biology*, Vol. I. New York: McGraw–Hill.

Gleason, J. R. 1998. dm64: Quantiles of the studentized range distribution. *Stata Technical Bulletin* 46: 6–10.

Snedecor, G. W. and W. G. Cochran. 1989. *Statistical Methods.* 7th ed. Ames, IA: Iowa State University Press.

sg102	Zero-truncated Poisson and negative binomial regression

Joseph Hilbe, Arizona State University, hilbe@asu.edu

Count response data is typically modeled using either Poisson or negative binomial regression. Zero counts are assumed to exist as a result of the distributional properties underlying both models. However, there are many count data situations which preclude the very possibility of zero counts; e.g., hospital length-of-stay data. In such data, counts begin with one, without the possibility of a count being equal to zero. When this occurs, the strict application of Poisson and negative binomial regression is inappropriate.

Poisson or negative binomial probability distributions that exclude zero do not sum to one. Therefore, an adjustment must be made to the underlying distributions upon which is based their respective log-likelihood functions. I shall describe the logic

of calculating the necessary adjustments and then present a brief example of the use of the programs `trpois0` and `trnbin0` which I have written that can be used to analyze zero-truncated regressions.

The Poisson probability distribution, where y is the random-response variable, can be expressed without subscripts as

$$\Pr(y;x) = \frac{\mu^y \exp(-\mu)}{y!}, \qquad y \geq 0$$

The Poisson probability of a zero count can likewise be expressed as

$$\Pr(y=0;x) = \exp(-\mu)$$

Hence the probability of a nonzero count is $1-\exp(-\mu)$. The probability of y, conditional upon $y>0$, is thus

$$\Pr(y|y>0;x) = \frac{\mu^y \exp(-\mu)}{y!(1-\exp(-\mu))}$$

The log-likelihood transformation for the above zero-truncated Poisson probability distribution is

$$\mathrm{LL}(\mu;x) = y\log(\mu) - \mu - \log\Gamma(y+1) - \log(1-\exp(-\mu))$$

In order to use Stata's `ml` command, the above log likelihood expression needs to be parameterized in terms of the linear predictor xB, that is, $\mu = \exp(xB)$. Hence, for the zero-truncated Poisson:

$$\mathrm{LL}(B;x) = yxB - \exp(xB) - \log\Gamma(y+1) - \log(1-\exp(-\exp(xB)))$$

Differentiation of this function provides the basis for calculating the robust "score":

$$y - \exp(xB) - \frac{\exp(xB)\exp(-\exp(xB))}{1-\exp(-\exp(xB))}$$

The log likelihood for the zero-truncated negative binomial may be determined using the same logic as above.

In this insert, I supply programs with `robust`, `cluster`, and `score` options and have provided the other standard Stata `ml` options. Fit, linear predictor, and residuals may be calculated afterward in the same manner as `poisson/poisml` and `nbreg/nbinreg`.

Long (1997) is an excellent source for discussion of these types of models.

Syntax

`trpois0` *depvar* [*varlist*] [*weight*] [`if` *exp*] [`in` *range*] [, `level(#)` `irr` `ltolerance(#)` `offset(`*varname*`)` `iterate(#)`]

`trnbin0` *depvar* [*varlist*] [*weight*] [`if` *exp*] [`in` *range*] [, `level(#)` `irr` `ltolerance(#)` `offset(`*varname*`)` `iterate(#)`]

`fweights` and `aweights` are allowed. Results may be redisplayed by issuing the command with no arguments or options.

Options

`level(#)` specifies the confidence level, in percent, for confidence intervals. The default is level(95) or as set by `set level`; see [U] **26.4 Specifying the width of confidence intervals**.

`irr` reports estimated coefficients transformed to incidence ratios, i.e., e^b rather than b. Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated. `irr` may be specified at estimation or when replaying previously estimated results.

`ltolerance(#)` specifies the maximum change in the log-likelihood function that can occur between iterations before convergence can be declared and defaults to 10^{-7}.

`offset(`*varname*`)` specifies a variable that is to be entered directly into the log-link function with coefficient constrained to be 1.

`iterate(#)` specifies the maximum number of iterations that are allowed before results are presented as if convergence had been achieved.

Example

This example dataset derives from HCFA's 1997 MedPar files. The MedPar consists of discharge abstracts from all Medicare patients who have been hospitalized within a given year. The data represents patients in the state of Arizona who were assigned a DRG (diagnostic related group) of 475 patients having a ventilator.

Length of stay (LOS) was shown to be better modeled using negative binomial regression than Poisson. Moreover, there does seem to be a substantial interhospital (`provnum`) correlation effect; hence the use of the `cluster` and `robust` options. Other possible predictors had previously been excluded after proving that they contributed little to the model.

The variables in the study are

`los`	length of stay
`died`	patient died while in hospital
`hmo`	patient a member of an HMO
`type1`	emergency admission
`type2`	urgent admission (first available bed)
`type3`	elective admission
`provnum`	hospital identifier

We begin by using negative binomial regression using the `nbinreg` command introduced in Hilbe (1998):

```
. nbinreg los died hmo type2-type3, nolog irr robust cluster(provnum)
Negative Binomial Estimates                         Number of obs    =    1495
                                                    Model chi2(4)    =  147.79
                                                    Prob > chi2      =  0.0000
Log Likelihood =  -4782.5988715                     Pseudo R2        =  0.0152
                          (standard errors adjusted for clustering on provnum)
------------------------------------------------------------------------------
             |               Robust
         los |        IRR   Std. Err.       z     P>|z|       [95% Conf. Interval]
-------------+----------------------------------------------------------------
los          |
        died |   .7890079   .0452559     -4.132   0.000        .705112    .8828859
         hmo |   .9318413   .0462707     -1.422   0.155       .8454256     1.02709
       type2 |   1.288153   .0818304      3.986   0.000       1.137351     1.45895
       type3 |    2.08867   .4417531      3.482   0.000       1.379879     3.16154
-------------+----------------------------------------------------------------
lnalpha      |
------------------------------------------------------------------------------
       alpha     .4352234   [lnalpha]_cons = ln(alpha)
                     (LR test against Poisson, chi2(1) =  4128.697 P =  0.0000)
```

Now we use `trnbin0`:

```
. trnbin0 los died hmo type2-type3, nolog irr robust cluster(provnum)
0-Truncated Negative Binomial Estimates             Number of obs    =    1495
                                                    Model chi2(4)    =  133.95
                                                    Prob > chi2      =  0.0000
Log Likelihood =  -4737.5350226                     Pseudo R2        =  0.0139
                          (standard errors adjusted for clustering on provnum)
------------------------------------------------------------------------------
             |               Robust
         los |        IRR   Std. Err.       z     P>|z|       [95% Conf. Interval]
-------------+----------------------------------------------------------------
los          |
        died |   .7770983   .0473858     -4.136   0.000       .6895592    .8757505
         hmo |   .9273564   .0493933     -1.416   0.157       .8354291    1.029399
       type2 |   1.308013   .0890172      3.945   0.000       1.144678    1.494655
       type3 |   2.152888   .4715409      3.501   0.000       1.401474    3.307179
-------------+----------------------------------------------------------------
lnalpha      |
------------------------------------------------------------------------------
       alpha     .5325344   [lnalpha]_cons = ln(alpha)
                     (LR test against Poisson, chi2(1) =  4218.824 P =  0.0000)
```

Since zero counts are not possible in this type of length-of-stay data, the latter model is most appropriate. Note that the log likelihood of the truncated model is less than that of the standard model.

References

Hilbe, J. 1998. sg91: Robust variance estimators for MLE Poisson and negative binomial regression. *Stata Technical Bulletin* 45: 26–28.

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

sg103	Within subjects (repeated measures) ANOVA, including between subjects factors

John R. Gleason, Syracuse University, loesljrg@ican.net

This insert presents `wsanova`, a command for within subjects or repeated measures analysis of variance (ANOVA). `wsanova` uses Stata's `anova` command to build a traditional ANOVA summary table for experimental designs having one within-subjects factor, and zero or more between-subjects factors; that is, `wsanova` handles both randomized blocks and split-plot factorial designs. If the design is unbalanced, the summary table presents Type III sums of squares and associated F tests. `wsanova` can also estimate the Greenhouse–Geisser degrees of freedom adjustment to compensate for violations of the sphericity condition assumed by F tests of within-subjects effects.

The syntax of the `wsanova` command is

`wsanova` *yvar wsfact* [*weight*] [`if` *exp*] [`in` *range*], `id(`*svar*`)` [`between(`*beffects*`)` `epsilon`

`nomatr` `wonly(`*weffects*`)`]

yvar names the response or dependent variable, and *wsfact* names the within-subject factor. `fweight`s and `aweight`s are allowed. The `id` option is not optional: *svar* is a variable that identifies the subjects; it must provide a unique number for each subject in the design. The `between` and `wonly` options are relevant only if there is at least one between-subjects factor; they are examined in connection with Examples 2 and 3. The `nomatr` option is ignored unless the `epsilon` option is present; the latter is explained in Examples 1–3, the former under *Saved results*, below.

Options

`id(`*svar*`)` declares that the variable *svar* uniquely identifies each subject to be used in the analysis.

`between(`*beffects*`)` supplies a list of between-subjects factors that classify the subjects, along with zero or more of their interactions. Up to seven such factors can be used; their interactions must be explicitly requested.

`wonly(`*weffects*`)` selects within-subjects effects that should be included in the analysis. By default, *wsfact* and all of its interactions with the elements of the `between` option are included.

`epsilon` requests that p values for within-subjects F tests be adjusted for lack of sphericity using the Greenhouse–Geisser and Huynh–Feldt adjustment factors. This option requires that *yvar* be nonmissing for each subject.

`nomatr` discards covariance and cell mean matrices at exit. The `epsilon` option creates a (pooled) covariance matrix `WSAoV_` and a matrix `WSAov_` of marginal means of *yvar* across levels of *wsfact*. In addition, when there are between subjects factors `epsilon` creates a cell means matrix and a within group covariance matrix for each distinct group of subjects; these matrices will be named `WSAov1` `WSAoV1`, `WSAov2` `WSAoV2`, ... By default, all these matrices are left in memory; `nomatr` erases each of them before `wsanova` exits.

Example 1: No between subjects factors

Consider a dataset presented by Cole and Grizzle (1966).

```
. use histamin, clear
(Blood histamine levels in dogs)
. describe
```

```
Contains data from histamin.dta
  obs:            64                          Blood histamine levels in dogs
 vars:             7                          8 Nov 1998 14:48
 size:         1,088 (99.7% of memory free)   * _dta has notes
-------------------------------------------------------------------------------
   1. group       byte   %8.0g       grp        Experimental group
   2. drug        byte   %8.0g       drug       Drug administered
   3. depleted    byte   %8.0g       nylbl      Depleted pre-test histamines?
   4. dog         byte   %8.0g                  Dog no.
   5. hist        float  %9.0g                  Histamine concentration
   6. time        byte   %8.0g                  Minutes after injection
   7. lhist       float  %9.0g                * log(Histamine concentration)
                                              * indicated variables have notes
-------------------------------------------------------------------------------
Sorted by:
```

Sixteen dogs were randomly divided into four experimental groups of equal size; dogs in each group were injected with a drug. The response variable is `lhist`, measured for each dog at `0`, `1`, `3`, and `5` minutes after injection of the drug. Isolating any given experimental group produces a single factor within subjects (randomized blocks) design, the default expectation of `wsanova`. For example:

```
. wsanova lhist time if group==1, id(dog)
                           Number of obs =      16     R-squared     =  0.9388
                           Root MSE      = .409681     Adj R-squared =  0.8979
                Source |  Partial SS    df       MS           F     Prob > F
           ------------+----------------------------------------------------
                   dog |  16.9024081     3  5.63413604
                  time |  6.25680792     3  2.08560264      12.43     0.0015
              Residual |  1.51054662     9  .167838513
           ------------+----------------------------------------------------
                 Total |  24.6697627    15  1.64465084
```

The usual F statistic finds strong evidence of differences in the mean of `lhist` across the four measurement times. Under the null hypothesis, this statistic is distributed as F with three and nine degrees of freedom assuming the usual sphericity condition (essentially, that the differences between all pairs of `lhist` means have the same variance).

Lack of sphericity tends to produce a positive bias in the F test for the within-subject factor (`time`); reducing the apparent degrees of freedom of the F test is a traditional way to reduce the bias. The main idea is to estimate an adjustment factor ε from the covariance matrix of the subjects' response vectors. This factor satisfies $1/(t-1) \leq \varepsilon \leq 1$, with $\varepsilon = 1$ if the sphericity assumption is satisfied; t is the number of levels of the within subjects factor ($t = 4$, in the present example). Multiplying the putative numerator and denominator degrees of freedom by the estimate of ε drives the p value upward (perhaps excessively so), thereby offsetting the positive bias.

The `epsilon` option computes the Greenhouse–Geisser ($\widehat{\varepsilon}$) and Huynh–Feldt ($\tilde{\varepsilon}$) estimates of ε. See Kirk (1995, Section 7.4) for additional details about these estimates. Adjusted p values are then reported along with the usual unadjusted p value. Repeating the example above gives

```
. wsanova lhist time if group==1, id(dog) eps
                           Number of obs =      16     R-squared     =  0.9388
                           Root MSE      = .409681     Adj R-squared =  0.8979
                Source |  Partial SS    df       MS           F     Prob > F
           ------------+----------------------------------------------------
                   dog |  16.9024081     3  5.63413604
                  time |  6.25680792     3  2.08560264      12.43     0.0015
              Residual |  1.51054662     9  .167838513
           ------------+----------------------------------------------------
                 Total |  24.6697627    15  1.64465084

     Note: Within subjects F-test(s) above assume sphericity of residuals;
           p-values corrected for lack of sphericity appear below.
Greenhouse-Geisser (G-G) epsilon: 0.4061
Huynh-Feldt (H-F) epsilon: 0.5376
                                          Sphericity      G-G        H-F
                Source |     df        F      Prob > F   Prob > F   Prob > F
           ------------+----------------------------------------------------
                  time |      3      12.43      0.0015     0.0267     0.0138
```

Thus, with the Greenhouse–Geisser estimate $\widehat{\varepsilon} = 0.4061$. We would approximate the sampling distribution of the F statistic by $F(1.22, 3.66)$, yielding less evidence ($p = 0.0267$) against the null hypothesis of no mean differences across `time` than the usual $p = 0.0015$ obtained from $F(3, 9)$.

Example 2: One between subjects factor

The data in `histamin.dta` arise from four experimental conditions (encoded by `group`) and one within-subjects factor (`time`); four independent layers of the layout considered in Example 1. Naming the between-subjects factor with the `between()` option produces a so-called "split-plot" ANOVA summary table:

```
. set matsize 42
. wsanova lhist time, id(dog) bet(group)
                          Number of obs =      63     R-squared     =  0.9706
                          Root MSE      = .267897     Adj R-squared =  0.9479
                  Source |  Partial SS    df       MS           F     Prob > F
              -----------+----------------------------------------------------
    Between subjects:    |  26.8507288     3  8.95024295       4.35     0.0271
                  group  |  26.8507288     3  8.95024295       4.35     0.0271
               dog*group |  24.6795159    12  2.05662633
                         |
     Within subjects:    |  31.2952849    12  2.60794041      36.34     0.0000
                    time |   12.771394     3  4.25713133      59.32     0.0000
              time*group |  18.5324756     9  2.05916396      28.69     0.0000
                Residual |  2.51191343    35  .071768955
              -----------+----------------------------------------------------
                   Total |  85.3495687    62  1.37660595
```

By default, the `between` option also includes in the model the interaction of the between- and within-subjects factors. This action can be suppressed with the `wonly` option; see Example 3, below. Note also that there is a single missing observation (for dog 6), so that the design is slightly unbalanced. The table above reports, without comment, what would elsewhere be called Type III sums of squares.

However, the `epsilon` option forces computation of covariance matrices, and that requires a complete data vector for each subject; i.e., no missing data. In this specific case, there should be little reason for concern about the within-subjects F tests but, as an example, excluding dog 6 we obtain

```
. set matsize 41
. wsanova lhist time if dog != 6, id(dog) bet(group) eps
                          Number of obs =      60     R-squared     =  0.9709
                          Root MSE      =  .27427     Adj R-squared =  0.9479
                  Source |  Partial SS    df       MS           F     Prob > F
              -----------+----------------------------------------------------
    Between subjects:    |  27.0286268     3  9.00954226       4.07     0.0359
                  group  |  27.0286268     3  9.00954226       4.07     0.0359
               dog*group |  24.3468341    11  2.21334855
                         |
     Within subjects:    |  31.3081774    12  2.60901478      34.68     0.0000
                    time |  12.0589871     3  4.01966235      53.44     0.0000
              time*group |  17.5232918     9  1.94703243      25.88     0.0000
                Residual |  2.48238892    33  .075223907
              -----------+----------------------------------------------------
                   Total |  85.1660271    59  1.44349199

      Note: Within subjects F-test(s) above assume sphericity of residuals;
            p-values corrected for lack of sphericity appear below.

Greenhouse-Geisser (G-G) epsilon: 0.5694
Huynh-Feldt (H-F) epsilon: 0.8475
                                          Sphericity      G-G        H-F
                  Source |     df        F      Prob > F   Prob > F   Prob > F
              -----------+----------------------------------------------------
                    time |      3      53.44      0.0000     0.0000     0.0000
              time*group |      9      25.88      0.0000     0.0000     0.0000
```

The `epsilon` option causes `wsanova` to compute $(p+1)$ covariance matrices of size $t \times t$, where p is the number of levels of the between-subjects factor and t is the number of levels of the within subjects factor. `wsanova` also computes $(p+1)$ matrices of size $1 \times t$ that hold the response variable means about which the covariance matrices were formed. Following the last call to `wsanova` above, there will be 10 such matrices:

```
. matrix dir
       WSAoV_[4,4]
       WSAoV4[4,4]
       WSAov_[1,4]
       WSAov4[1,4]
       WSAoV3[4,4]
       WSAov3[1,4]
       WSAoV2[4,4]
       WSAov2[1,4]
       WSAoV1[4,4]
       WSAov1[1,4]
```

See *Saved results* (below) for details of the naming convention. By way of illustration,

```
. matrix list WSAov1

WSAov1[1,4]
                0           1           3           5
Means  -2.8905289  -1.1621239  -1.9995999  -2.3229961

. matrix list WSAoV_

symmetric WSAoV_[4,4]
           0          1          3          5
0  .36349405
1   .3682182  .67015436
3  .47351738  .69791689    .812182
5  .40802547  .57215993   .6873491  .59318986
```

displays the means of `lhist` across the four levels of `time` within the first experimental group, and the pooled covariance matrix from which $\widehat{\varepsilon} = 0.5694$ and $\tilde{\varepsilon} = 0.8475$ were computed. The column names are the levels of the within-subjects factor (`time`).

Example 3: Two or more between subjects factors

The four experimental groups in our running example correspond to a 2×2 factorial design: two kinds of drug crossed with two levels of histamine at injection time:

```
. tab drug depleted

Drug       | Depleted pre-test histamines?
administere|
d          |        No        Yes |     Total
-----------+----------------------+----------
  Morphine |        16         16 |        32
   TriMeth |        16         16 |        32
-----------+----------------------+----------
     Total |        32         32 |        64
```

Thus, the analysis could be performed in terms of the factors `drug` and `depleted` rather than in terms of the single factor `group`. There is of course no fundamental advantage in doing so; it is merely traditional to request a model with main effects and interactions:

```
. set matsize 61
. wsanova lhist time if dog != 6, id(dog) bet(drug depl drug*depl)
                          Number of obs =      60     R-squared     =  0.9709
                          Root MSE      =  .27427     Adj R-squared =  0.9479

                  Source |  Partial SS    df       MS           F     Prob > F
              -----------+----------------------------------------------------
    Between subjects:    |  27.0286268     3  9.00954226       4.07     0.0359
                  drug   |  5.99336256     1  5.99336256       2.71     0.1281
              depleted   |  15.4484076     1  15.4484076       6.98     0.0229
          drug*depleted  |  4.69087549     1  4.69087549       2.12     0.1734
       dog*drug*depleted |  24.3468341    11  2.21334855
                         |
     Within subjects:    |  31.3081774    12  2.60901478      34.68     0.0000
                  time   |  12.0589871     3  4.01966235      53.44     0.0000
             time*drug   |  1.84429539     3  .614765129       8.17     0.0003
          time*depleted  |  12.0897855     3  4.02992849      53.57     0.0000
      time*drug*depleted |  2.93077944     3  .976926479      12.99     0.0000
              Residual   |  2.48238892    33  .075223907
              -----------+----------------------------------------------------
                   Total |  85.1660271    59  1.44349199
```

All between-subjects effects must be explicitly named in the `between()` option, but all interactions of those named effects with the within-subjects factor are, by default, included automatically. If this is undesirable, the `wonly()` option can be used to list those within-subjects effects that *should* be included in the model. For example, suppose we want an analysis that excludes the `drug*depleted` interaction from the between-subjects effects, includes as within-subjects effects only the `time` and `time*depleted` interaction, and adjusts for lack of sphericity:

```
. set matsize 33
. wsanova lhist time if dog != 6, id(dog) bet(drug depl) wonly(time time*depl)
> eps
                            Number of obs =      60     R-squared     =  0.9103
                            Root MSE      = .442692     Adj R-squared =  0.8642

                   Source |  Partial SS    df       MS           F     Prob > F
                ----------+----------------------------------------------------
    Between subjects:     |  22.3377513     2  11.1688756       4.62     0.0326
                     drug |  6.87754936     1  6.87754936       2.84     0.1176
                 depleted |  16.8857304     1  16.8857304       6.98     0.0215
       dog*drug*depleted  |  29.0377096    12  2.41980913
                          |
     Within subjects:     |  26.1474934     6  4.35791556      22.24     0.0000
                     time |  12.0454347     3   4.0151449      20.49     0.0000
           time*depleted  |  12.3626079     3  4.12086929      21.03     0.0000
                 Residual |  7.64307289    39  .195976228
                ----------+----------------------------------------------------
                    Total |  85.1660271    59  1.44349199

      Note: Within subjects F-test(s) above assume sphericity of residuals;
            p-values corrected for lack of sphericity appear below.
Greenhouse-Geisser (G-G) epsilon: 0.5694
Huynh-Feldt (H-F) epsilon: 0.7651
                                            Sphericity      G-G        H-F
                   Source |     df        F      Prob > F   Prob > F   Prob > F
                ----------+----------------------------------------------------
                     time |      3      20.49      0.0000     0.0000     0.0000
             time*depleted |      3      21.03      0.0000     0.0000     0.0000
```

At this point, '`matrix dir`' will produce exactly the same response as in Example 2, and the 10 matrices will be identical to those in Example 2. That is, a covariance matrix and a matrix of cell means is computed for each distinct combination of the between-subjects factors, regardless of whether all possible between-subjects effects have been included. That is why $\widehat{\varepsilon} = 0.5694$ here, exactly the same as for the second model fit in Example 2. (But $\tilde{\varepsilon}$ differs from Example 2 because it depends on the between-subjects error degrees of freedom.)

Saved results

At exit, the contents of `_result()` will be exactly as though the command '`test` *weffect*' had been issued, where *weffect* corresponds to the final F statistic in the ANOVA summary table. In particular, the within-subjects error (Residual) sum of squares and *df* will be in `_result(4)` and `_result(5)`, respectively. `wsanova` also saves in the global macro `WSAoVcmd` a parsed copy of the command line that launched it.

The `epsilon()` option saves results as follows. Suppose that the argument of the `between()` option contains factors named `A`, `B`, `C`,..., that there are p usable combinations of the levels of those factors, and that the within-subjects factor has t levels. `epsilon()` causes `wsanova` to compute a $t \times t$ within-group covariance matrix from each of those p subgroups of observations. The matrices will be named `WSAoV1`, `WSAoV2`,..., `WSAoV`p, with the final character indexing the `ABC`... combinations in lexicographic order. In addition, `wsanova` computes p matrices of size $1 \times t$ of associated cell means, named `WSAov1`, `WSAov2`, ..., `WSAov`p. Finally, `wsanova` computes a $t \times t$ matrix `WSAoV_` that pools the covariance matrices `WSAoV1`, `WSAoV2`, ..., `WSAoV`p, and a $1 \times t$ matrix of marginal means, named `WSAov_`. If the `between()` option is absent, only the matrices `WSAoV_` and `WSAov_` are created. For each covariance matrix it creates, `wsanova` stores $\widehat{\varepsilon}$ in global macro `WSAoV_E`. The final entry is $\widehat{\varepsilon}$ for the pooled covariance matrix `WSAoV_`; the value reported by the `epsilon` option. Global macro `WSAoV_df` contains the degrees of freedom for each of the covariance matrices, ordered to match the entries of `WSAoV_E`.

By default, all the `WSAoV`$*$ and `WSAov`$*$ matrices are left behind when `wsanova` exits. They will be automatically erased the next time `wsanova` is called with the `epsilon` option, but the `nomatr` option causes `wsanova` to instead discard all these matrices before it exits.

Remarks

1. `wsanova` builds its ANOVA summary table by calling the `anova` command. Consequently, `predict` can be used to obtain predicted values and residuals following `wsanova`, exactly as after `anova`.

2. Thus, `wsanova` makes the same demands as `anova` on the `matsize` parameter, and that can be problematic. Because of the way `anova` handles factors, the minimum matrix size can be surprisingly large; larger than the number of available observations, as in the first model fit in Example 3. Moreover, the minimum `matsize` depends heavily on how one chooses to represent the between-subjects effects, if any. (`anova` is most wasteful when there are many factors with just two levels.) In the examples above, the `set matsize` commands always show the minimum setting for the `wsanova command` that follows.

3. The `between()` option will permit up to seven between-subjects factors and their various interactions; this is an upper limit imposed by Stata's `test` command. Only a single within-subjects factor is permitted. This may seem restrictive, but multiple within-subjects factors can be converted to a single factor, just as `group` represents the two factors `drug` and `depleted` in Examples 2 and 3, above. Indeed, some would argue that in repeated measures designs there is but one true within-subjects factor; the sequence in which the repeated observations were made.

Acknowledgment

This project was supported by a grant R01-MH54929 from the National Institute on Mental Health to Michael P. Carey.

References

Cole, J. W. L. and J. E. Grizzle. 1966. Applications of multivariate analysis of variance to repeated measures experiments. *Biometrics* 22: 810–828.

Kirk, R. E. 1995. *Experimental Design: Procedures for the Behavioral Sciences*. 3rd ed. Belmont, CA: Brooks/Cole Publishing.

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 (409–845–3144) 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
Address:	P.O. Box 1169 17100 NAZERATH-ELLIT Israel
Phone:	+972 (0)6 6100101
Fax:	+972 (0)6 6554254
Email:	assc@netvision.net.il
Countries served:	Israel

Company:	Axon Technology Company Ltd
Address:	9F, No. 259, Sec. 2 Ho-Ping East Road TAIPEI 106 Taiwan
Phone:	+886-(0)2-27045535
Fax:	+886-(0)2-27541785
Email:	hank@axon.axon.com.tw
Countries served:	Taiwan

Company:	Chips Electronics
Address:	Lokasari Plaza 1st Floor Room 82 Jalan Mangga Besar Raya No. 82 JAKARTA Indonesia
Phone:	62 - 21 - 600 66 47
Fax:	62 - 21 - 600 66 47
Email:	puyuh23@indo.net.id
Countries served:	Indonesia

Company:	Dittrich & Partner Consulting
Address:	Kieler Strasse 17 5. floor D-42697 Solingen Germany
Phone:	+49 2 12 / 26 066 - 0
Fax:	+49 2 12 / 26 066 - 66
Email:	sales@dpc.de
URL:	http://www.dpc.de
Countries served:	Germany, Austria, Italy

Company:	IEM
Address:	P.O. Box 2222 PRIMROSE 1416 South Africa
Phone:	+27-11-8286169
Fax:	+27-11-8221377
Email:	iem@hot.co.za
Countries served:	South Africa, Botswana, Lesotho, Namibia, Mozambique, Swaziland, Zimbabwe

Company:	MercoStat Consultores
Address:	9 de junio 1389 CP 11400 MONTEVIDEO Uruguay
Phone:	598-2-613-7905
Fax:	Same
Email:	mercost@adinet.com.uy
Countries served:	Uruguay, Argentina, Brazil, Paraguay

Company:	Metrika Consulting
Address:	Mosstorpsvagen 48 183 30 Taby STOCKHOLM Sweden
Phone:	+46-708-163128
Fax:	+46-8-7924747
Email:	sales@metrika.se
Countries served:	Sweden, Baltic States, Denmark, Finland, Iceland, Norway

Company:	Ritme Informatique
Address:	34, boulevard Haussmann 75009 Paris France
Phone:	+33 (0)1 42 46 00 42 +33 (0)1 42 46 00 33
Email:	info@ritme.com
URL:	http://www.ritme.com
Countries served:	France, Belgium, Luxembourg

(*List continued on next page*)

International Stata Distributors

(*Continued from previous page*)

Company:	Scientific Solutions S.A.
Address:	Avenue du Général Guisan, 5 CH-1009 Pully/Lausanne Switzerland
Phone:	41 (0)21 711 15 20
Fax:	41 (0)21 711 15 21
Email:	info@scientific-solutions.ch
Countries served:	Switzerland

Company:	Smit Consult
Address:	Doormanstraat 19 5151 GM Drunen Netherlands
Phone:	+31 416-378 125
Fax:	+31 416-378 385
Email:	J.A.C.M.Smit@smitcon.nl
URL:	http://www.smitconsult.nl
Countries served:	Netherlands

Company:	Survey Design & Analysis Services P/L
Address:	249 Eramosa Road West Moorooduc VIC 3933 Australia
Phone:	+61 (0)3 5978 8329
Fax:	+61 (0)3 5978 8623
Email:	sales@survey-design.com.au
URL:	http://survey-design.com.au
Countries served:	Australia, New Zealand

Company:	Timberlake Consultants
Address:	47 Hartfield Crescent WEST WICKHAM Kent BR4 9DW United Kingdom
Phone:	+44 (0)181 462 0495
Fax:	+44 (0)181 462 0493
Email:	info@timberlake.co.uk
URL:	http://www.timberlake.co.uk
Countries served:	United Kingdom, Eire

Company:	Timberlake Consulting S.L.
Address:	Calle Mendez Nunez, 1, 3 41011 Sevilla Spain
Phone:	+34 (9) 5 422 0648
Fax:	+34 (9) 5 422 0648
Email:	timberlake@zoom.es
Countries served:	Spain

Company:	Timberlake Consultores, Lda.
Address:	Praceta Raúl Brandao, n°1, 1°E 2720 ALFRAGIDE Portugal
Phone:	+351 (0)1 471 73 47
Fax:	+351 (0)1 471 73 47
Email:	timberlake.co@mail.telepac.pt
Countries served:	Portugal

Company:	Unidost A.S.
	Rihtim Cad. Polat Han D:38
	Kadikoy
	81320 ISTANBUL
	Turkey
Phone:	+90 (216) 414 19 58
Fax:	+30 (216) 336 89 23
Email:	info@unidost.com
URL:	http://abone.turk.net/unidost
Countries served:	Turkey

Company:	Vishvas Marketing-Mix Services
Address:	"Prashant" Vishnu Nagar Baji Prabhu Deshpande Path, Naupada THANE - 400602 India
Phone:	+91-251-440087
Fax:	+91-22-5378552
Email:	vishvas@vsnl.com
Countries served:	India