STATA TECHNICAL BULLETIN

May 1999
STB-49

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.

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

an69	STB-43–STB-48 available in bound format

Patricia Branton, Stata Corporation, stata@stata.com

The eighth year of the *Stata Technical Bulletin* (issues 43–48) has been reprinted in a bound book called *The Stata Technical Bulletin Reprints, Volume 8*. The volume of reprints is available from StataCorp for $25, plus shipping. Authors of inserts in STB-43–STB-48 will automatically receive the book at no charge and need not order.

This book of reprints includes everything that appeared in issues 43–48 of the STB. As a consequence, you do not need to purchase the reprints if you saved your STBs. However, many subscribers find the reprints useful since they are bound in a convenient volume. Our primary reason for reprinting the STB, though, is to make it easier and cheaper for new users to obtain back issues. For those not purchasing the *Reprints*, note that *zz9* in this issue provides a cumulative index for the eighth year of the original STBs.

dm45.1	Changing string variables to numeric: update

Nicholas J. Cox, University of Durham, UK, n.j.cox@durham.ac.uk

Syntax

`destring` [*varlist*] [`, noconvert noencode float`]

Remarks

`destring` was published in STB-37. Please see Cox and Gould (1997) for a full explanation and discussion. It is here translated into the idioms of Stata 6.0. The main substantive change is that because value labels may now be as long as 80 characters, string variables of any length, from `str1` to `str80`, may be encoded to numeric variables with string labels.

Reference

Cox, N. J. and W. Gould. 1997. dm45: Changing string variables to numeric. *Stata Technical Bulletin* 37: 4–6. Reprinted in *Stata Technical Bulletin Reprints*, vol. 7, pp. 34–37.

dm65	A program for saving a model fit as a dataset

Roger Newson, Imperial College School of Medicine, London, UK, r.newson@ic.ac.uk

The command `parmest` is designed to save a model fit in a data set, either in memory, or on disk, or both. It was inspired by the example of `collapse`. It takes, as input, the parameter estimates of the most recently fitted model, and their covariance matrix. It creates, as output, a new dataset, with one observation per parameter, and variables corresponding to equation names (if present), parameter names, estimates, standard errors, z or t test statistics, p-values and confidence limits. This output dataset may be saved to a disk file, or remain in memory (overwriting the pre-existing dataset), or both.

Typically, `parmest` is used with `graph` to produce confidence interval plots. It is also possible to sort the output dataset by p-value, in order to carry out closed test procedures, like those of Holm, Hommel, or Holland and Copenhaver, summarized in Wright (1992).

Syntax

`parmest` [`, dof(#) label eform level(#) fast saving(`*filename*[`,replace`]`) norestore`]

Options

`dof(#)` specifies the degrees of freedom for t-distribution-based confidence limits. If `dof` is zero, then confidence limits are calculated using the standard normal distribution. If `dof` is absent, it is set to a default according to the last estimation results.

`label` indicates that a variable named *label* is to be generated in the new dataset, containing the variable labels of variables corresponding to the parameter names, if such variables can be found in the existing dataset.

`eform` indicates that the estimates and confidence limits are to be exponentiated, and the standard errors multiplied by the exponentiated estimates.

`level(#)` specifies the confidence level, in percent, for confidence limits. The default is `level(95)` or as set by `set level`. (See **[U] Estimation and post-estimation commands**.)

`fast` specifies that `parmest` not go to extra work so that it can restore the original data should the user press *Break*. `fast` is intended for use by programmers.

`saving(`*filename*`[,replace])` saves the output dataset in a file. If `replace` is specified, and a file of name *filename* already exists, then the old file is overwritten.

`norestore` specifies whether or not the pre-existing dataset is restored at the end of execution. This option is automatically set to `norestore` if `fast` is specified or `saving(`*filename*`)` is absent, otherwise it defaults to restoring the pre-existing dataset.

Remarks

`parmest` creates a new dataset with one observation per parameter and data on the most recent model fit. There are two character variables, `eq` and `parm`, containing equation and parameter names, respectively. The numeric variables are `estimate`, `stderr`, `z` (or `t`), `p`, `min`xx and `max`xx, where xx is the value of the `level` option. These variables contain parameter estimates, standard errors, z test (or t test) statistics, p-values, and confidence limits, respectively. The p-values test the hypothesis that the appropriate parameter is zero, or one if `eform` is specified.

Example

This example uses the Stata example dataset `auto.dta`, with the added variable `manuf`, containing the first word of `make`, and denoting manufacturer. (See **[U] 26.10 Obtaining robust variance estimates** for an example of the use of this variable.) We want to derive confidence intervals for the average fuel efficiency (in miles per gallon) for each manufacturer, using a homoscedastic regression model. (Some manufacturers are represented by only one model in the dataset, so their specific variances cannot be estimated.) We then want to plot the confidence intervals by manufacturer.

We proceed as follows. First we tabulate `manuf`, generating the dummy variables for the regression analysis:

```
. tabulate manuf,missing gene(manu)
Manufacturer|      Freq.     Percent        Cum.
------------+-----------------------------------
         AMC |          3        4.05        4.05
        Audi |          2        2.70        6.76
         BMW |          1        1.35        8.11
       Buick |          7        9.46       17.57
         Cad. |          3        4.05       21.62
        Chev. |          6        8.11       29.73
       Datsun |          4        5.41       35.14
        Dodge |          4        5.41       40.54
         Fiat |          1        1.35       41.89
         Ford |          2        2.70       44.59
        Honda |          2        2.70       47.30
        Linc. |          3        4.05       51.35
        Mazda |          1        1.35       52.70
        Merc. |          6        8.11       60.81
         Olds |          7        9.46       70.27
      Peugeot |          1        1.35       71.62
        Plym. |          5        6.76       78.38
        Pont. |          6        8.11       86.49
      Renault |          1        1.35       87.84
       Subaru |          1        1.35       89.19
       Toyota |          3        4.05       93.24
           VW |          4        5.41       98.65
        Volvo |          1        1.35      100.00
------------+-----------------------------------
        Total |         74      100.00
```

We then carry out a regression analysis of `mpg` with respect to the dummy variables:

```
. regress mpg manu1-manu23, noconst
  Source |       SS       df       MS              Number of obs =      74
---------+------------------------------           F( 23,    51) =   70.51
   Model |  34910.1286    23  1517.83168           Prob > F      =  0.0000
Residual |  1097.87143    51  21.5268908           R-squared     =  0.9695
---------+------------------------------           Adj R-squared =  0.9558
   Total |    36008.00    74  486.594595           Root MSE      =  4.6397
```

```
------------------------------------------------------------------------------
     mpg |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
   manu1 |   20.33333   2.678737      7.591   0.000       14.95555    25.71112
   manu2 |         20   3.280769      6.096   0.000       13.41358    26.58642
   manu3 |         25   4.639708      5.388   0.000        15.6854     34.3146
   manu4 |   19.14286   1.753645     10.916   0.000       15.62227    22.66345
   manu5 |   16.33333   2.678737      6.097   0.000       10.95555    21.71112
   manu6 |         22   1.894153     11.615   0.000       18.19733    25.80267
   manu7 |      25.75   2.319854     11.100   0.000        21.0927     30.4073
   manu8 |      20.25   2.319854      8.729   0.000        15.5927     24.9073
   manu9 |         21   4.639708      4.526   0.000        11.6854     30.3146
  manu10 |       24.5   3.280769      7.468   0.000       17.91358    31.08642
  manu11 |       26.5   3.280769      8.077   0.000       19.91358    33.08642
  manu12 |   12.66667   2.678737      4.729   0.000       7.288878    18.04445
  manu13 |         30   4.639708      6.466   0.000        20.6854     39.3146
  manu14 |   17.16667   1.894153      9.063   0.000         13.364    20.96934
  manu15 |   19.42857   1.753645     11.079   0.000       15.90798    22.94916
  manu16 |         14   4.639708      3.017   0.004       4.685398     23.3146
  manu17 |       26.2   2.074941     12.627   0.000       22.03438    30.36562
  manu18 |       19.5   1.894153     10.295   0.000       15.69733    23.30267
  manu19 |         26   4.639708      5.604   0.000        16.6854     35.3146
  manu20 |         35   4.639708      7.544   0.000        25.6854     44.3146
  manu21 |   22.33333   2.678737      8.337   0.000       16.95555    27.71112
  manu22 |       28.5   2.319854     12.285   0.000        23.8427     33.1573
  manu23 |         17   4.639708      3.664   0.001       7.685398     26.3146
------------------------------------------------------------------------------
```

We then use `parmest` to save the parameter estimates, and their confidence limits, to the new dataset.

```
. parmest,lab
. describe
Contains data
  obs:            23
 vars:             8
 size:         1,656 (82.7% of memory free)
------------------------------------------------------------------------------
   1. parm        str6   %9s                 Parameter name
   2. label       str14  %14s                Parameter label
   3. estimate    double %10.0g              Parameter estimate
   4. stderr      double %10.0g              SE of parameter estimate
   5. t           double %10.0g              t-test statistic
   6. p           double %10.0g              P-value
   7. min95       double %10.0g              Lower 95% confidence limit
   8. max95       double %10.0g              Upper 95% confidence limit
------------------------------------------------------------------------------
Sorted by:
     Note:  data has changed since last save
. list parm label estimate stderr
          parm           label    estimate      stderr
  1.     manu1      manuf==AMC   20.333333   2.6787367
  2.     manu2     manuf==Audi          20    3.280769
  3.     manu3      manuf==BMW          25    4.639708
  4.     manu4    manuf==Buick   19.142857   1.7536448
  5.     manu5     manuf==Cad.   16.333333   2.6787367
  6.     manu6    manuf==Chev.          22   1.8941529
  7.     manu7   manuf==Datsun       25.75    2.319854
  8.     manu8    manuf==Dodge       20.25    2.319854
  9.     manu9     manuf==Fiat          21    4.639708
 10.    manu10     manuf==Ford        24.5    3.280769
 11.    manu11    manuf==Honda        26.5    3.280769
 12.    manu12    manuf==Linc.   12.666667   2.6787367
 13.    manu13    manuf==Mazda          30    4.639708
 14.    manu14    manuf==Merc.   17.166667   1.8941529
 15.    manu15     manuf==Olds   19.428571   1.7536448
 16.    manu16  manuf==Peugeot          14    4.639708
 17.    manu17    manuf==Plym.        26.2   2.0749405
 18.    manu18    manuf==Pont.        19.5   1.8941529
 19.    manu19  manuf==Renault          26    4.639708
 20.    manu20   manuf==Subaru          35    4.639708
 21.    manu21   manuf==Toyota   22.333333   2.6787367
 22.    manu22       manuf==VW        28.5    2.319854
 23.    manu23    manuf==Volvo          17    4.639708
```

```
. list parm estimate min95 max95 t p
          parm    estimate       min95       max95           t           p
  1.     manu1   20.333333   14.955545   25.711122   7.5906428   6.372e-10
  2.     manu2          20   13.413581   26.586419   6.0961317   1.450e-07
  3.     manu3          25   15.685398   34.314602   5.3882701   1.830e-06
  4.     manu4   19.142857   15.622268   22.663446    10.91604   5.972e-15
  5.     manu5   16.333333   10.955545   21.711122   6.0974016   1.443e-07
  6.     manu6          22    18.19733    25.80267   11.614691   6.151e-16
  7.     manu7       25.75   21.092699   30.407301   11.099836   3.265e-15
  8.     manu8       20.25   15.592699   24.907301   8.7289975   1.074e-11
  9.     manu9          21   11.685398   30.314602   4.5261469   .00003625
 10.    manu10        24.5   17.913581   31.086419   7.4677613   9.947e-10
 11.    manu11        26.5   19.913581   33.086419   8.0773745   1.100e-10
 12.    manu12   12.666667   7.2888785   18.044455   4.7285971   .00001823
 13.    manu13          30   20.685398   39.314602   6.4659241   3.796e-08
 14.    manu14   17.166667   13.363996   20.969337   9.0629784   3.306e-12
 15.    manu15   19.428571   15.907983    22.94916   11.078966   3.496e-15
 16.    manu16          14   4.6853976   23.314602   3.0174312   .00397119
 17.    manu17        26.2   22.034383   30.365617   12.626868   2.553e-17
 18.    manu18        19.5    15.69733    23.30267    10.29484   4.745e-14
 19.    manu19          26   16.685398   35.314602   5.6038009   8.504e-07
 20.    manu20          35   25.685398   44.314602   7.5435781   7.557e-10
 21.    manu21   22.333333   16.955545   27.711122   8.3372634   4.332e-11
 22.    manu22        28.5   23.842699   33.157301   12.285256   7.363e-17
 23.    manu23          17   7.6853976   26.314602   3.6640236   .00059118
```

We then augment this new dataset with two new variables, the character variable `manufb` and the numeric variable `manufn`, derived from the variable labels stored in `label`, and representing the first two letters of the manufacturer's name. Finally, we use `manufn` to create a confidence interval plot for mean fuel efficiencies by manufacturer:

```
. gene str2 manufb=substr(label,length("manuf==")+1,2)
. encode manufb,gene(manufn)
. set textsize 100
. graph estimate min95 max95 manufn,c(.II) s(O..)
>  xscale(0.5,23.5)
>  xlabel(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23)
>  yscale(0,45) ylabel(0,5,10,15,20,25,30,35,40,45)
>  t1title(" ") t2title(" ")
>  b2title("Manufacturer") l2title("Mileage (miles per gallon)")
>  saving(fig1.gph,replace);
```

The graph generated by this program is given as Figure 1.

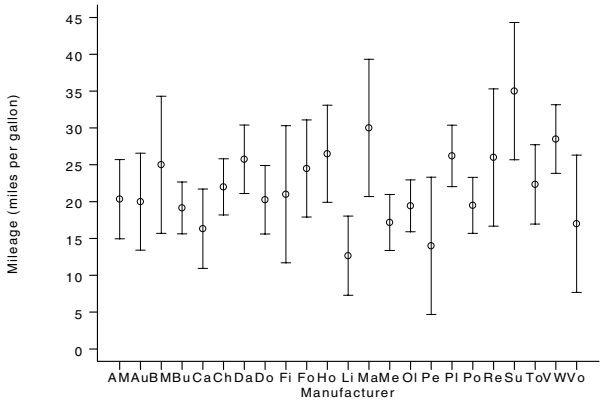

Figure 1. Confidence interval plot for mean fuel efficiencies by manufacturer.

Acknowledgments

I would like to thank Nick Cox of Durham University, UK, Jonah B. Gelbach at the University of Maryland at College Park, and Phil Ryan at the Department of Public Health, University of Adelaide, Australia for giving many helpful suggestions for improvements on previous versions posted to Statalist.

Reference

Wright, S. P. 1992. Adjusted p-values for simultaneous inference. *Biometrics* 48: 1005–1013.

dm66	Recoding variables using grouped values

David Clayton, MRC Biostatistical Research Unit, Cambridge, david.clayton@mrc-bsu.cam.ac.uk
Michael Hills (retired), mhills@regress.demon.co.uk

This insert describes a new option in `egen` which creates a new categorical variable from a metric variable. The categorical variable is coded with either the left-hand ends of the grouping intervals specified, or the integer codes 0, 1, 2, etc. The integer codes can be labeled with the left-hand ends of the intervals. If no intervals are specified, the command creates k groups for which the frequency of observations are approximately equal. Missing values are ignored when counting the frequencies.

Syntax

`egen` *newvar* `= cut(`*varname*`), { breaks(#,#,...,#) | group(#) } [ icodes label ]`

Options

`breaks(#,#,...,#)` supplies the breaks for the groups, in ascending order. The list of break points may be simply a list of numbers separated by commas, but can also include the syntax `a[b]c`, meaning from `a` to `c` in steps of size `b`. If no breaks are specified, the command expects the option `group()`.

`group(#)` specifies the number of equal frequency grouping intervals to be used in the absence of `breaks`. Specifying this option automatically invokes `icodes`.

`icodes` requests that the codes 0, 1, 2, etc. be used in place of the left-hand ends of the intervals.

`label` requests that the integer coded values of the grouped variable be labeled with the left-hand ends of the grouping intervals. Specifying this option automatically invokes `icodes`.

Example

Using the variable `length` from the `auto` data, the commands

```
. egen lgrp = cut(length), breaks(140,180,200,220,240)
. tab lgrp
```

produce the output

```
       lgrp |      Freq.     Percent        Cum.
------------+-----------------------------------
        140 |         31       41.89       41.89
        180 |         16       21.62       63.51
        200 |         20       27.03       90.54
        220 |          7        9.46      100.00
------------+-----------------------------------
      Total |         74      100.00
```

as will the command

```
. egen lgrp = cut(length), breaks(140,180[20]240)
```

Values outside the range 140–240 are coded as missing. The command

```
. egen lgrp = cut(length), breaks(140,180[20]240) icodes
```

will produce a variable coded 0, 1, 2, 3, and adding the option `label` will label the integer coded values of the grouped variable with the labels 140–, 180–, 200–, 220–. Finally the commands

```
. egen lgrp = cut(length), group(5) label
. tab lgrp
```

will produce the output

```
       lgrp |      Freq.     Percent        Cum.
------------+-----------------------------------
       142- |         12       16.22       16.22
       165- |         16       21.62       37.84
       179- |         14       18.92       56.76
       198- |         15       20.27       77.03
       206- |         17       22.97      100.00
------------+-----------------------------------
      Total |         74      100.00
```

The algorithm for producing equal frequency groups is to first use the Stata command `pctile` to calculate the quantiles, and then to use these together with the extreme values of the variable being cut, as breaks. The result is groups of approximately equal frequency with the additional property that duplicate observations must all lie in the same group.

Discussion

Some of these results could be obtained using the Stata commands `summarize`, `pctile` and `xtile`. For example,

```
. summarize length
. pctile pct = length, nq(5)
. xtile lgrp = length, cut(pct)
```

is equivalent to

```
. egen lgrp = cut(length), group(5)
```

but the `cut` option in `egen` puts everything in the same table. Theoretically, `xtile` could be used to reproduce the results from

```
. egen lgrp = cut(length), breaks(140,180,200,220,240)
```

but in practice this would be cumbersome, because the breaks need to be in a variable. The Stata function `recode()` is also a candidate, but now the grouped categorical variable is coded with the right-hand ends. In spite of overlap with these existing commands, it seems to us that there is room for a new one which combines all the common requirements when categorizing a metric variable in a simple way.

dm67 | Numbers of missing and present values

Nicholas J. Cox, University of Durham, UK, n.j.cox@durham.ac.uk

Syntax

`nmissing` [*varlist*] [`if` *exp*] [`in` *range*] [`, min(#)`]

`npresent` [*varlist*] [`if` *exp*] [`in` *range*] [`, min(#)`]

Description

`nmissing` lists the number of missing values in each variable in *varlist*. Missing means `.` for numeric variables and the empty string `""` for string variables.

`npresent` lists the number of present (nonmissing) values in each variable in *varlist*.

Options

`min(#)` specifies that only numbers at least # should be listed. The default is one.

Remarks

Suppose you want a concise report on the numbers of missing values in a large dataset. You are interested in string variables as well as numeric variables. Existing Stata commands do not serve this need. `summarize` is biased towards numeric variables and reports all string variables as having 0 observations, meaning 0 observations that can be treated as numeric. `inspect` has the same bias, and in any case has no concise mode. `codebook` comes nearer, in that strings are treated as strings and not as failed numeric variables, but it again has no concise mode.

`nmissing` is an attempt to fill this gap. When called with no arguments it reports on the whole dataset, including both numeric and string variables. If a *varlist* is specified, or the minimum number of values to be reported is specified by the `min( )` option, then the focus is restricted accordingly.

`npresent` is the complementary command that reports on present (nonmissing) values. `nmissing` and `npresent` are written for Stata 6.0.

The user-written command `pattern` (Goldstein 1996a, 1996b) may also be useful in this connection. It reports, as the name implies, on the pattern of missing data for one or more variables.

Examples

With the familiar auto dataset,

```
. nmissing
```

yields

```
rep78             5
```

while

```
. nmissing if foreign
```

yields

```
rep78             1
```

References

Goldstein, R. 1996a. sed10: Patterns of missing data. *Stata Technical Bulletin* 32: 12–13. Reprinted in *Stata Technical Bulletin Reprints*, vol. 6, p. 115.

——. 1996b. sed10.1: Update to pattern. *Stata Technical Bulletin* 33: 2. Reprinted in *Stata Technical Bulletin Reprints*, vol. 6, pp. 115–116.

gr34.2	Drawing Venn diagrams

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

The Venn diagram routine has been updated to allow more than 32,767 observations. An error in the previous version found by Steven Stillman has been corrected. The error made the contents of a generated variable faulty, in particular with missing data. The counts in the actual Venn Diagram Graph have been correct in previous versions.

References

Lauritsen, J. M. 1999a. gr34: Drawing Venn diagrams. *Stata Technical Bulletin* 47: 3–8.

——. 1999b. gr34.1: Drawing Venn diagrams. *Stata Technical Bulletin* 48: 2.

gr36	An extension of for, useful for graphics commands

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

Arguably, one of the most useful and powerful features of Stata is the `for` command that allows the simple programming of the repetition of commands with somewhat different arguments. However, for graphics commands I find the `for` command somewhat inconvenient; rather than inspecting the graphs one at a time, I want to look at a single *combined plot* to facilitate comparison of the plots. To make this easier, I wrote `forgraph` which is actually just a slight modification of the `for` command. To look at histograms for a number of variables from the Stata automobile data, one can issue the command

```
forgraph price-hdroom: graph @, hist xlab ylab
```

which gives Figure 1.

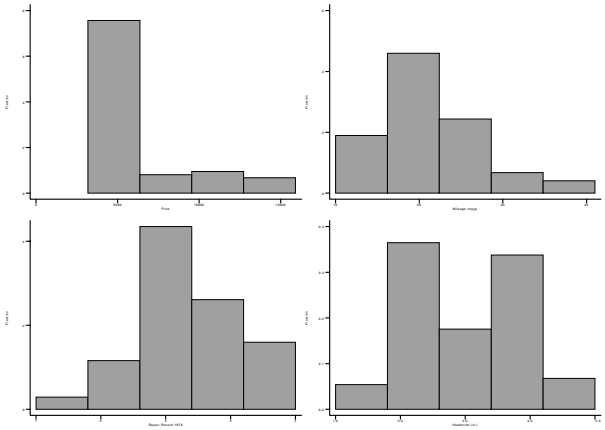

Figure 1. Using forgraph to obtain four histograms.

`forgraph` works with other graphics commands as well. To obtain a plot for kernel density estimates of these variables one can use the command

```
forgraph price-hdroom: kdensity @,
```

which gives Figure 2.

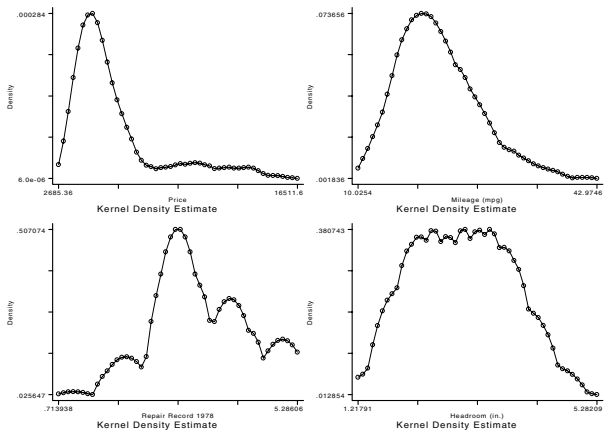

Figure 2. Four kernel density estimates.

Note that the phrase "Kernel estimates" is displayed by `kdensity` in each plot. This looks rather ugly. Also the labels are not quite readable. We may improve the quality of the plot as follows

```
forgraph price-hdroom, margin(10) title(Kernel estimates) tsize(200): kdensity @, ti(".") xlab ylab
```

which gives Figure 3.

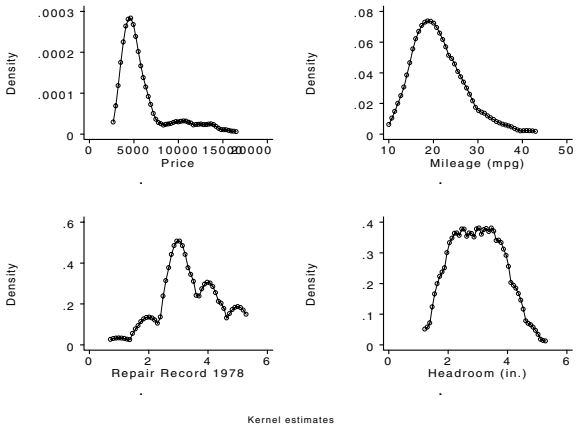

Figure 3. A more readable version of Figure 2.

`forgraph` has options `margin`, `title`, and `tsize` to specify the width between the subplots in the combined plot, the title for the combined plot, and the textsize used in the subplots. Finally, `forgraph` supports an option `saving` to save the combined plot as a `gph` file.

Syntax

`forgraph` *list* [, `title`(*str*) `margin`(*#*) `tsize`(*#*) `saving`(*filename*) *for_options*] : *graphics_cmd*

Example

A last illustration of `forgraph` demonstrates how it can be used to prepare graphs separately for subgroups of the data. Stata's default display for `twoway` plots with the `by` option is particularly attractive. Also, some of Stata's graphics commands do not support the `by` option. To illustrate, we do a scatterplot of `price` versus `mpg` highlighting the first four types of foreign cars:

```
. sort rep78
. gen rep781 = rep78
```

```
. replace rep781 = . if rep78==5
. hilite price mpg, hilite(foreign) gap(4) ylab by(rep781) border saving(forgraph4, replace)
```

which gives Figure 4.

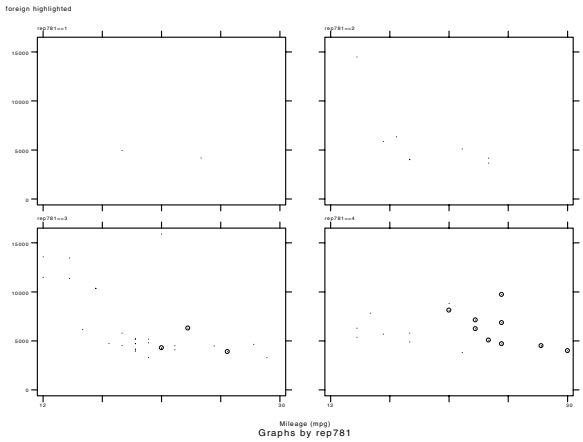

Figure 4. Using the `hilite` command.

Then

```
. forgraph 1-4, lt(num) ti(foreign cars highlighted) mar(10) ts(200):
> hilite price mpg if rep78==@, hilite(foreign) gap(4) ylab border t1(repair record @)
```

which gives Figure 5.

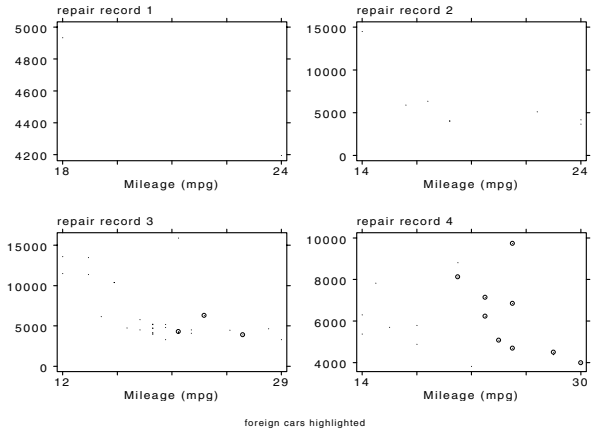

Figure 5. Using `hilite` and `forgraph`

Remark

When I decided to write a special version of `for` for graphics commands, I thought about extending the `for` command with an option `graph` and the other options that I added in `forgraph`. The reason is, simply, that I am somewhat scared by the proliferation of variants of standard Stata commands that add relatively minor functionality, or package combinations of standard Stata commands. When StataCorp publishes an updated version of the standard command, the variant becomes outdated. Clearly, I would much welcome that StataCorp would include my graphics extension in `for` in a future release. But, maybe it is more important that StataCorp works at modifying the Stata system to support object-oriented programming so that a user command can *inherit* all properties of parent commands. This, I realize, would not be a trivial piece of work for StataCorp, but it will make Stata easier to maintain in the long run.

gr37	Cumulative distribution function plots

David Clayton, MRC Biostatistical Research Unit, Cambridge, david.clayton@mrc-bsu.cam.ac.uk
Michael Hills (retired), mhills@regress.demon.co.uk

A plot of the empirical cumulative distribution function of a variable is a convenient way of looking at the empirical distribution without having to choose bins, as in histograms. The Stata command `cumul` is rather primitive, and a new command

cdf is offered as an alternative. With cdf, distributions can be compared within subgroups defined by a second variable, and the best fitting normal (Gaussian) model can be superimposed over the empirical cdf.

Syntax

cdf *varname* [*weight*] [if *exp*] [in *range*] [, by(*varname*) normal samesd *graph_options*]

aweights, fweights, iweights, and pweights are allowed.

Options

by(*varname*) causes a separate cdf to be calculated for each value of *varname*, on the same graph.

normal causes a normal probability curve with the same mean and standard deviation to be superimposed over the cdf.

samesd is relevant only when by and normal options are used together. It fits normal curves with different means but the same standard deviations, demonstrating the fit of the Gaussian location shift model.

graph_options are allowed. Default labeling is supplied when *graph_options* are absent, but the x-axis label may be supplied in the b2 graphics option and the y-axis may be labeled using the l1 option. If the xlog option is used, the normal option causes log normal distributions to be fitted.

Examples

The data refer to numbers of t4 cells in blood samples from 20 patients in remission from Hodgkin's disease and 20 patients in remission from disseminated malignancies. They are taken from Practical Statistics for Medical Research by Altman (see Shapiro et al. 1986). The two variables are t4 for the count and grp, coded 1 or 2. The command

```
. cdf t4, by(grp) xlab ylab
```

produces the graph in Figure 1. The second cdf has been leaned on relative to the first which suggests using the log T4 cell count.

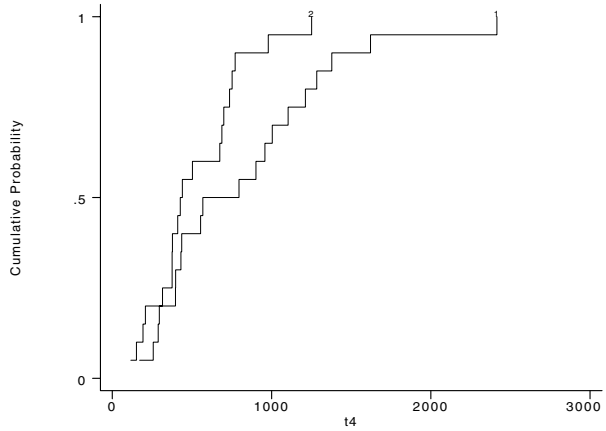

Figure 1. cdf for t4 cell counts for two types of patients

The commands

```
. gen logt4=log(t4)
. cdf logt4, by(grp) xlab ylab
```

produce the graph in Figure 2, while

```
. cdf logt4, by(grp) normal same xlab ylab
```

gives Figure 3.

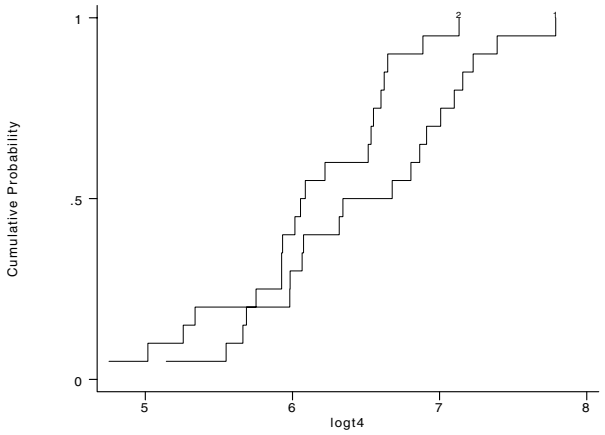

Figure 2. cdf for logarithm of t4 cell counts

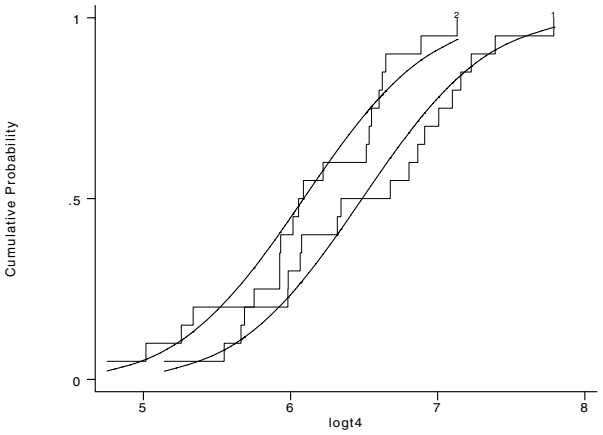

Figure 3. Figure 2 with Gaussian cdfs superimposed

Reference

Shapiro et al. 1986. Practical Statistics for Medical Research. *American Journal of Medical Science* 293: 366–370.

sbe27	Assessing confounding effects in epidemiological studies

Zhiqiang Wang, Menzies School of Health Research, Darwin, Australia, wang@menzies.su.edu.au

In epidemiological studies, investigators sometimes lack prior knowledge about whether a covariate is a confounder and thus employ a strategy that uses the data to help them decide whether to adjust for a variable (Maldonado and Greenland 1993). With the change-in-estimate approach, a variable is selected for control only if its control seems to make a substantial difference in the exposure effect estimates. Depending on the study design and characteristics of the data, we may use logistic regressions, Poisson regressions, or Cox proportional hazard models to estimate the effect of exposure and to adjust for confounding. The effect estimates (EE) can be odds ratio (OR), rate ratio (RR) or hazard ratio (HR). In this insert we present the command `epiconf` which calculates and graphs adjusted effect measures such as OR, RR and HR and their confidence intervals. It also calculates change-in-estimates after adding a potential confounder into the model with the forward selection approach or deleting a potential confounder from the model with the backward deletion approach. The order of variables being selected is based on the magnitude of the change-in-estimate.

`epiconf` uses either a forward selection or backward deletion method. The forward selection method starts from the crude estimate without adjusting for any confounder. Then `epiconf` adds the confounders for adjustment one-by-one in a stepwise fashion, at each step adding the covariate with the largest change-in-estimate. The backward deletion method starts with the estimate adjusted for all potential confounders. Then `epiconf` deletes the confounders from adjustment one-by-one in a stepwise fashion, at each step deleting the covariate with the least change-in-estimate. `epiconf` also reports p-values from the Wald type collapsibility test statistic: significance-test-of-the-change (Maldonado and Greenland 1993):

$$\text{Change-in-estimate}(\%) = \begin{cases} \dfrac{\text{EE}_{\text{adj.x}} - \text{EE}_{\text{unadj.x}}}{\text{EE}_{\text{unadj.x}}} \times 100\%, & \text{forward selection method} \\ \dfrac{\text{EE}_{\text{unadj.x}} - \text{EE}_{\text{adj.x}}}{\text{EE}_{\text{adj.x}}} \times 100\%, & \text{backward deletion method} \end{cases}$$

The exact cut-point for importance is somewhat arbitrary and may vary from study to study. `epiconf` provides crude, all adjusted effect estimates and change-in-estimates, which allows investigators to chose an appropriate cut-point for their own studies. Maldonado and Greenland (1993) suggested that the change-in-estimate method performed best when the cut-point for deciding whether adjusted and unadjusted estimates differ by an important amount was set to a low value (10%). A higher than conventional α level should be considered when we use the significance-test-of-the-change (0.20). Our decision about importance could also be influenced by the method (forward or backward) we choose, as shown by the example given below. A more detailed discussion on selecting confounders can be found in Rothman and Greenland (1998).

Syntax

`epiconf` *yvar xvar* [`if` *exp*] [`in` *range*] [`, con(`*covarlist*`) cat(`*covarlist*`) model(logit|poisson|cox)`

`expos(`*var*`) dead(`*var*`) detail nograph backward coeff level(`#`)` *graph_options*]

where *yvar* is a binary outcome variable for logistic or Poisson regression, or a survival time variable for the Cox proportional hazards model. *xvar* is a binary exposure variable of interest.

Options

`con(`*covarlist*`)` specifies continuous potential confounding variables.

`cat(`*covarlist*`)` specifies nominal potential confounding variables.

`model(logit|poisson|cox)` specifies the regression method. The default is `logit`.

`expos(`*varname*`)` specifies a variable that reflects the amount of exposure over which the *yvar* events were observed for each observation. This option is only for Poisson regression.

`dead(`*varname*`)` specifies the name of a variable recording 0 if censored and nonzero (typically 1) if failure. If `dead()` is not specified, all observations are assumed to have failed. This option is only for Cox regression.

`detail` gives details at each step. The default is a summary.

`nograph` yields no graph.

`backward` specifies the selection strategy as the backward deletion method. The default is the forward selection method.

`coef` graphs regression coefficients instead of effect estimates.

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

Examples

We use a dataset (included on the accompanying diskette) providing information on association between albuminuria and risk of death in a particular population. To assess confounding effects, we use Poisson regressions in `epiconf`.

```
. use conf
. describe
Contains data from conf.dta
  obs:           743
 vars:             9                          22 Nov 1998 21:51
 size:        32,692 (95.4% of memory free)
-------------------------------------------------------------------------------
   1. dead        float  %9.0g                 Death
   2. ab_uria     float  %9.0g                 Albuminuria
   3. age         long   %12.0g                Age in years
   4. sex         float  %9.0g       sex       SEX
   5. hypert      float  %9.0g                 Hypertension
   6. hichol      float  %9.0g                 High cholesterol
   7. weight      float  %9.0g                 Body weight, kg
   8. smoke       float  %9.0g                 smoking
   9. time        double %10.0g                observed time
-------------------------------------------------------------------------------
```

First we use forward selection:

```
. epiconf dead ab_uria, con(age weight) cat(hich hyper smoke sex) model(poisson) expos(time)
Assessment of Confounding Effects Using Change-in-Estimate Method
------------------------------------------------------------
  Outcome:    "dead"
  Exposure:   "ab_uria"
  N =          743
------------------------------------------------------------
Forward approach
Potential confounders were added one at a time sequentially
------------+-----------------------------------------------
            |                        Change in Rate ratio
 Adj Var    | Rate Ratio   95% CI   ---------------------
            |                          %        p>|z|
------------+-----------------------------------------------
      Crude |   3.26   2.01,  5.26        .         .
       +age |   1.98   1.21,  3.23    -39.3    0.00000
    +weight |   2.22   1.34,  3.68     12.0    0.07291
   +i.smoke |   1.94   1.15,  3.25    -12.6    0.01962
  +i.hichol |   2.05   1.21,  3.48      6.1    0.20995
  +i.hypert |   1.94   1.14,  3.30     -5.5    0.06328
    +i.sex* |   1.96   1.15,  3.34      0.7    0.80716
------------+-----------------------------------------------
*Adjusted for all potential confounders
```

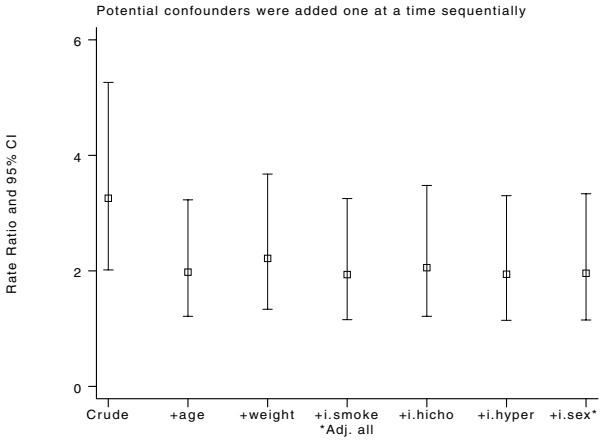

Figure 1. The result of using forward selection.

Note that the rate ratios in the above output and figure from a forward selection method are rate ratios adjusted for the corresponding variable plus all previous variable(s) if any. Nominal variables are labeled as `i.varname`. We see that `age` is an important confounder that is the first to be adjusted for. Adding age into the model makes a substantial change (39.3%) in the rate ratio estimate. After the age confounding effect has been adjusted for, the rate ratio only changes slightly by adjusting for other variables. If we take 10% as a cut-point of importance, we need to adjust for age, weight and smoking. The adjusted rate ratio is 1.94 with a 95% confidence interval of (1.15, 3.25). If we take 20% as the cut-point of importance, we need only adjust for age. The adjusted rate ratio is 1.98 with 95% confidence interval (1.21, 3.23).

Next we use the backward deletion method:

```
. epiconf dead ab_uria, con(age weight) cat(hich hyper smoke sex)
> model(poisson) expos(time) backward
Assessment of Confounding Effects Using Change-in-Estimate Method
------------------------------------------------------------
  Outcome:    "dead"
  Exposure:   "ab_uria"
  N =          743
------------------------------------------------------------
Backward approach
Potential confounders were removed one at a time sequentially
------------+-----------------------------------------------
            |                          Change in Rate ratio
 Adj Var    | Rate Ratio   95% CI   ---------------------
            |                            %        p>|z|
------------+-----------------------------------------------
   Adj. all |   1.85   1.09,  3.13        .         .
     -i.sex |   1.83   1.09,  3.09     -0.9    0.77795
  -i.hypert |   1.94   1.15,  3.25      5.7    0.11986
    -weight |   1.78   1.08,  2.93     -8.1    0.22131
   -i.smoke |   1.98   1.21,  3.23     11.2    0.03345
      -age* |   3.26   2.01,  5.26     64.6    0.00000
------------+-----------------------------------------------
*Crude estimate
```

(*Graph on next page*)

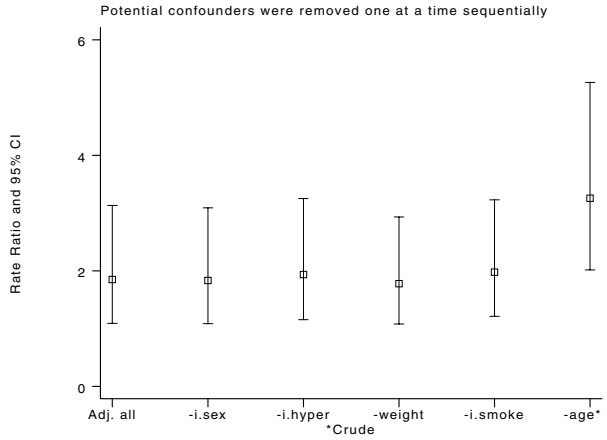

Figure 2. The result of using backward deletion.

With a backward deletion method, the rate ratio adjusted for all variables (`Adj. all`) is presented first. Then, `epiconf` deletes the nominal variable `sex` first because deleting it makes the least change-in-estimate (0.9%). The most important confounder (age) in terms of change in estimate is the last covariate to be deleted. If we take 10% as a cut-point of importance, we need adjust for age and smoking. The adjusted rate ratio is 1.78 with 95% confidence interval (1.08, 2.93), while if we take 20% as a cut-point of importance, we need only adjust for age. The adjusted rate ratio is 1.98 with a 95% confidence interval (1.21, 3.23).

Acknowledgment

I thank Nicholas Cox for providing a subroutine `vallist` and Jean Bouyer for useful suggestions.

References

Maldonado, G. and S. Greenland. 1993. Simulation study of confounder-selection strategies. *American Journal of Epidemiology* 138: 923–936.

Rothman, K. J. and S. Greenland. 1998. *Modern Epidemiology*. Philadelphia: Lippincott–Raven.

sbe28	Meta-analysis of p-values

Aurelio Tobias, Statistical Consultant, Madrid, Spain, bledatobias@ctv.es

Fisher's work on combining of p-values (Fisher 1932) has been suggested as the origin of meta-analysis (Jones 1995). However, combination of p-values presents serious disadvantages, relative to combining estimates. For example, when p-values are testing different null hypotheses, they do not consider the direction of the association combining opposing effects, they cannot quantify the magnitude of the association, nor study heterogeneity between studies. Combination of p-values may be the only available option if nonparametric analyses of individual studies have been performed or if little information apart from the p-value is available about the result of a particular study (Jones 1995).

Fisher's method

This method (Fisher 1932) combines the probabilities of several hypotheses tests, testing the same null hypothesis

$$U = -2\sum_{j=1}^{k} \ln(p_j)$$

where the p_j are the one-tailed p-values for each study, and k is the number of studies. Then U follows a χ^2 distribution with $2k$ degrees of freedom. This method is not suggested to combine a large number of studies because it tends to reject the null hypothesis routinely (Rosenthal 1984). It also tends to have problems combining studies that are statistically significant, but in opposite directions (Rosenthal 1980).

Edgington's methods

The first method (Edgington 1972a) is based on the sum of probabilities

$$p = \left(\sum_{j=1}^{K} p_j\right)^k \Big/ k!$$

The results obtained are similar to Fisher's method, but it is also restricted for a small number of studies. This method presents problems when the sum of probabilities is higher than one; in this situation the combined probability tends to be conservative (Rosenthal 1980).

An alternative method was also suggested by Edgington (1972b), to combine more than four studies, based on the contrast of the p-value average

$$\overline{p} = \sum_{j=1}^{k} p_j \Big/ k$$

in which case $U = (0.5 - \overline{p})\sqrt{12}$ follows a normal distribution.

Syntax

The command `metap` works on a dataset containing the p-values for each study. The syntax is as follows:

`metap` *pvar* [`if` *exp*] [`in` *range*] [`, e(#)`]

Options

`e(#)` combines the p-values using Edgington's methods. Here, two alternatives are available; specifying `a` means that the additive method based on the sum of probabilities is used, while `n` specifies that the normal curve method based on the contrast of the p-value average is used. By default, Fisher's method is used.

Example

We consider data from seven placebo-controlled studies on the effect of aspirin in preventing death after myocardial infarction. Fleiss (1993) published an overview of these data. Let us assume that each study included in the meta-analysis is testing the same null hypothesis $H_0 : \theta \leq 0$ versus the alternative $H_1 : \theta > 0$. If the estimate of the log odds ratio and its standard error is available, then one-tailed p-values can easily be generated using the `normprob` function:

```
. generate pvar=normprob(-logrr/logse)
. list studyid logrr logse pvar, noobs
 studyid     logrr     logse       pvar
   MCR-1    0.3289    0.1972   .0476728
     CDP    0.3853    0.2029   .0287845
   MRC-2    0.2192    0.1432   .0629185
    GASP    0.2229    0.2545   .1905599
   PARIS    0.2261    0.1876   .1140584
    AMIS   -0.1249    0.0981   .8985248
  ISIS-2    0.1112    0.0388   .0020786
```

In this situation, all methods to combine p-values produce similar results:

```
. metap pvar
Meta-analysis of p_values
------------------------------------------------------------
 Method             |   chi2         p_value      studies
--------------------+---------------------------------------
 Fisher             |   38.938235    .00037283    7
------------------------------------------------------------

. metap pvar, e(a)
Meta-analysis of p_values
------------------------------------------------------------
 Method             |   .            p_value      studies
--------------------+---------------------------------------
 Edgington, additive|   .            .00157658    7
------------------------------------------------------------

. metap pvar, e(n)
Meta-analysis of p_values
------------------------------------------------------------
 Method             |   Z            p_value      studies
--------------------+---------------------------------------
 Edgington, Normal  |   2.8220842    .00238563    7
------------------------------------------------------------
```

These figures agree with the result obtained using the `meta` command introduced in Sharp and Sterne (1998) on a fixed effects ($z = 3.289$, $p = 0.001$) and random effects ($z = 2.093$, $p = 0.036$) models, respectively. However, the combination of p-values presents the serious limitations described previously.

Individual or frequency records

As for other meta-analysis commands, `metap` works on data contained in frequency records, one for each study or trial.

Saved results

`metap` saves the following results:

`S_1`	Method used to combine the p-values
`S_2`	number of studies
`S_3`	Statistic used to obtain the combined probability
`S_4`	Values of the statistic described in `S_3`
`S_5`	Combined probability

References

Edgington, E. S. 1972a. An additive method for combining probability values from independent experiments. *Journal of Psychology* 80: 351–363.

——. 1972b. A normal curve method for combining probability values from independent experiments. *Journal of Psychology* 82: 85–89.

Fisher, R. A. 1932. *Statistical Methods for Research Workers*. 4th ed. London: Oliver & Boyd.

Fleiss, J. L. 1993. The statistical basis of meta-analysis. *Statistical Methods in Medical Research* 2: 121–149.

Jones, D. 1995. Meta-analysis: weighing the evidence. *Stat Med* 14: 137–149.

Rosenthal, R. (Ed.) 1980. *New Directions for Methodology of Social and Behavioral Science*. Vol. V. San Francisco: Sage.

Rosenthal, R. 1984. Valid interpretation of quantitative research results. In *New Directions for Methodology of Social and Behavioral Science: Forms of Validity in Research, 12*, ed. D. Brinberg and L. Kidder. San Francisco: Jossey–Bass.

Sharp, S. and J. Sterne. 1998. sbe16.1: New syntax and output for the meta-analysis command. *Stata Technical Bulletin* 42: 6–8.

sg64.1 | Update to pwcorrs

Fred Wolfe, Arthritis Research Center, Wichita, KS, fwolfe@southwind.net

This update corrects a problem in `pwcorrs`, see Wolfe (1997). When the option `vars()` was not specified and `bonferroni` or `sidak` was specified, the program reported p-values of 0.0000 instead of the correct values.

Reference

Wolfe, F. 1997. sg64: pwcorrs: An enhanced correlation display. *Stata Technical Bulletin* 35: 22–25. Reprinted in *Stata Technical Bulletin Reprints*, vol. 6, pp. 163–167.

sg81.1 | Multivariable fractional polynomials: update

Patrick Royston, Imperial College School of Medicine, UK, proyston@rpms.ac.uk
Gareth Ambler, Imperial College School of Medicine, UK, gambler@rpms.ac.uk

Introduction

Multivariable fractional polynomials (FPs) were introduced by Royston & Altman (1994) and implemented in a command `mfracpol` for Stata 5 by Royston and Ambler (1998). The model selection procedure in the Stata 5 version was essentially the backward elimination algorithm described by Royston and Altman (1994) with modifications described by Sauerbrei and Royston (1999) (see the technical note below). An application of multivariable FPs in modeling prognostic and diagnostic factors in breast cancer is given by Sauerbrei and Royston (1999) (see our example below).

Briefly, fractional polynomial models are especially useful when one wishes to preserve the continuous nature of the predictor variables in a regression model, but suspects that some or all the relationships may be nonlinear. Using a backfitting algorithm, `mfracpol` finds a fractional polynomial transformation for each continuous predictor, fixing the current functional forms of the other predictor variables. The algorithm terminates when the functional forms of the predictors do not change.

Commands `stfracp` and `stmfracp` implementing respectively univariate and multivariable FPs for the survival (`st`) data format were presented by Royston (1998).

The present insert has two main purposes:

1. To update `mfracpol`, `stfracp` and `stmfracp` for Stata 6.
2. To describe improved FP model selection algorithms in `mfracpol`.

We have kept the same name (`mfracpol`) for the multivariable FP command.

The syntax of `stfracp` and `stmfracp` is unchanged, except that both programs inherit the rich set of options available with `stcox` in Stata 6. The syntax of `mfracpol` is basically as described by Royston & Ambler (1998). Changes are summarized below.

Changes to mfracpol

The main differences between the previous and new versions of `mfracpol` are as follows:

1. The new version is compatible only with Stata 6. It does not work with Stata 5.
2. The default model selection algorithm has been changed.
3. The new options: `adjust()`, `dfdefault()`, `sequential`, `xorder()`, `xpowers()` are available.
4. FPs of degree higher than 2 are supported via the `df()` and `dfdefault()` options.
5. The default operation of the `df()` option has been altered.
6. The screen display of the convergence process of the algorithm has been altered.
7. New variables created by `mfracpol` are named according to the conventions used by `fracpoly`.

Syntax of the mfracpol command

See the help file for full details. The default degrees of freedom (df) for a predictor are assigned by the `df()` option according to the number of distinct (unique) values of the predictor as shown in the following table.

No. of distinct values	**Default df**
1	(invalid, must be >1)
2–3	1 (straight line model)
4–5	min(2, `dfdefault(#)`)
≥ 6	`dfdefault(#)`

`dfdefault(#)` determines the default maximum df for a predictor, the default # being 4 (second degree FP). The `adjust()` option works in the same way as the `adjust()` option in `fracpoly`. The default is `adjust(mean)` unless the predictor is binary, in which case adjustment is to the lower of the two distinct values. The `xorder(`*order*`)` option allows you to change the ordering of covariates presented to the selection algorithm. *order* may be `+` (the default, with the most significant predictor in a multiple linear regression model taken first), `-` (reverse of `+`, with the least significant predictor taken first) or `n` (no ordering, i.e., the predictors are taken in the order specified by *xvarlist*). The `xpowers()` option allows you to specify customized powers for any subset of the continuous predictors.

Example

We illustrate two of the analyses performed by Sauerbrei and Royston (1999). We use `brcancer.dta` which contains prognostic factors data from the German Breast Cancer Study Group of patients with node-positive breast cancer. The dataset was downloaded in text form from web site `http://www.blackwellpublishers.co.uk/rss/`. The response variable is recurrence-free survival time (`rectime`) and the censoring variable is `censrec`. There are 686 patients with 299 events. We use Cox regression to predict the log hazard of recurrence from prognostic factors of which 5 are continuous (`x1`, `x3`, `x5`, `x6`, `x7`) and 3 are binary (`x2`, `x4a`, `x4b`). Hormonal therapy (`hormon`) is known to reduce recurrence rates and is forced into the model. We use `mfracpol` to build a model from the initial set of 8 predictors using the backfitting model selection algorithm. We set the nominal p-value for variable and for FP selection to 0.05 for all variables except `hormon` for which it is set to 1:

```
. mfracpol cox rectime x1 x2 x3 x4a x4b x5 x6 x7 hormon, dead(censrec)
> alpha(.05) select(.05, hormon:1)

Deviance for model with all terms untransformed =  3471.637, 686 observations

Variable Model (vs.)   Deviance  Dev diff.   P      Powers    (vs.)
-------------------------------------------------------------------
x5       m=2   (null)  3503.610    61.366  0.000    .5 3      .
               (lin.)  3471.637    29.393  0.000    .5 3      1
               (m=1)   3449.203     6.959  0.031    .5 3      0
x5       final         3442.244                     .5 3

x6       m=2   (null)  3464.113    29.917  0.000    -2 .5     .
               (lin.)  3442.244     8.048  0.045    -2 .5     1
               (m=1)   3435.550     1.354  0.508    -2 .5     .5
x6       final         3435.550                     .5

[hormon included with 1 df in model]

x4a      lin.  (null)  3440.749     5.199  0.023    1         .
x4a      final         3435.550                     1

x3       m=2   (null)  3436.832     3.560  0.469    -2 3      .
x3       final         3436.832                     .

x2       lin.  (null)  3437.589     0.756  0.384    1         .
x2       final         3437.589                     .

x4b      lin.  (null)  3437.848     0.259  0.611    1         .
x4b      final         3437.848                     .

x1       m=2   (null)  3437.893    18.085  0.001    -2 -.5    .
               (lin.)  3437.848    18.040  0.000    -2 -.5    1
               (m=1)   3433.628    13.820  0.001    -2 -.5    -2
x1       final         3419.808                     -2 -.5

x7       m=2   (null)  3420.805     3.715  0.446    -.5 3     .
x7       final         3420.805                     .
-------------------------------------------------------------------
Cycle 1: deviance =    3420.805
-------------------------------------------------------------------

x5       m=2   (null)  3494.867    74.143  0.000    -2 -1     .
               (lin.)  3451.795    31.071  0.000    -2 -1     1
               (m=1)   3428.023     7.299  0.026    -2 -1     0
x5       final         3420.724                     -2 -1

x6       m=2   (null)  3452.093    32.704  0.000    0 0       .
               (lin.)  3427.703     8.313  0.040    0 0       1
               (m=1)   3420.724     1.334  0.513    0 0       .5
x6       final         3420.724                     .5

[hormon included with 1 df in model]

x4a      lin.  (null)  3425.310     4.586  0.032    1         .
x4a      final         3420.724                     1

x3       m=2   (null)  3420.724     5.305  0.257    -.5 0     .
x3       final         3420.724                     .

x2       lin.  (null)  3420.724     0.214  0.644    1         .
x2       final         3420.724                     .

x4b      lin.  (null)  3420.724     0.145  0.703    1         .
x4b      final         3420.724                     .

x1       m=2   (null)  3440.057    19.333  0.001    -2 -.5    .
               (lin.)  3440.038    19.314  0.000    -2 -.5    1
               (m=1)   3436.949    16.225  0.000    -2 -.5    -2
x1       final         3420.724                     -2 -.5

x7       m=2   (null)  3420.724     2.152  0.708    -1 3      .
x7       final         3420.724                     .

Fractional polynomial fitting algorithm converged after 2 cycles.

Transformations of covariates:

-> gen double Ix1__1 = X^-2-.0355 if e(sample)
-> gen double Ix1__2 = X^-.5-.4342 if e(sample)
   (where: X = x1/10)
-> gen double Ix5__1 = X^-2-3.984 if e(sample)
-> gen double Ix5__2 = X^-1-1.996 if e(sample)
   (where: X = x5/10)
-> gen double Ix6__1 = X^.5-.3332 if e(sample)
   (where: X = (x6+1)/1000)
```

```
Final multivariable fractional polynomial model for rectime
-------------------------------------------------------------
Variable |     -----Initial-----          -----Final-----
         |   df     Select   Alpha    Status    df    Powers
---------+---------------------------------------------------
      x1 |    4     0.0500   0.0500     in      4     -2 -.5
      x2 |    1     0.0500   0.0500     out     0
      x3 |    4     0.0500   0.0500     out     0
     x4a |    1     0.0500   0.0500     in      1     1
     x4b |    1     0.0500   0.0500     out     0
      x5 |    4     0.0500   0.0500     in      4     -2 -1
      x6 |    4     0.0500   0.0500     in      2     .5
      x7 |    4     0.0500   0.0500     out     0
  hormon |    1     1.0000   0.0500     in      1     1
---------+---------------------------------------------------

Cox regression -- Breslow method for ties
Entry time 0                                    Number of obs   =        686
                                                LR chi2(7)      =     155.62
                                                Prob > chi2     =     0.0000
Log likelihood = -1710.3619                     Pseudo R2       =     0.0435
------------------------------------------------------------------------------
 rectime |
 censrec |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
  Ix1__1 |   44.73377   8.256682      5.418   0.000       28.55097    60.91657
  Ix1__2 |  -17.92302   3.909611     -4.584   0.000      -25.58571   -10.26032
     x4a |   .5006982   .2496324      2.006   0.045       .0114276    .9899687
  Ix5__1 |   .0387904   .0076972      5.040   0.000       .0237041    .0538767
  Ix5__2 |  -.5490645   .0864255     -6.353   0.000      -.7184554   -.3796736
  Ix6__1 |  -1.806966   .3506314     -5.153   0.000      -2.494191   -1.119741
  hormon |  -.4024169   .1280843     -3.142   0.002      -.6534575   -.1513763
------------------------------------------------------------------------------
Deviance: 3420.724.
```

Some explanation of the output from the model selection algorithm is desirable. Consider the first few lines of output in the iteration log:

```
1. Deviance for model with all terms untransformed = 3471.637, 686 observations
   Variable Model (vs.)   Deviance  Dev diff.   P      Powers    (vs.)
   ------------------------------------------------------------------------
2. x5       m=2   (null)  3503.610    61.366  0.000    .5 3      .
3.               (lin.)  3471.637    29.393  0.000    .5 3      1
4.               (m=1)   3449.203     6.959  0.031    .5 3      0
5. x5       final         3442.244                     .5 3
```

Line 1 gives the deviance ($-2 *$ log partial likelihood) for the Cox model with all terms linear, showing where the algorithm starts. The model is modified variable-by-variable in subsequent steps. The most significant linear term turns out to be `x5` which is therefore processed first. Line 2 compares the best-fitting FP with $m = 2$ for `x5` with a model omitting `x5`. The FP has powers (0.5,3) and the test for inclusion of `x5` is highly significant. The reported deviance of 3503.610 is for the null model, not for the model with $m = 2$. The deviance for the $m = 2$ model may be calculated by subtracting the deviance difference (`Dev diff.`) from the reported deviance, giving $3503.610 - 61.366 = 3442.244$. Line 3 shows that the $m = 2$ model is also a highly significantly better fit than a straight line (`lin.`) and line 4 that it is also somewhat better than an FP with $m = 1$ ($P = 0.031$). Thus at this stage in the model selection procedure the final model for `x5` (line 5) is an FP with powers (0.5,3). The overall model with $m = 2$ for `x5` and all other terms linear has deviance 3442.244.

After all the variables have been processed (cycle 1) and reprocessed (cycle 2) in this way, convergence is achieved since the functional forms (FP powers and variables included) after cycle 2 are the same as after cycle 1. The model finally chosen is Model II as given in Tables 3 and 4 of Sauerbrei and Royston (1999). Due to scaling of variables, the regression coefficients reported there are different, but the model and its deviance are identical. It includes `x1` with powers $(-2, -0.5)$, `x4a`, `x5` with powers $(-2, -1)$, and `x6` with power 0.5. There is strong evidence of nonlinearity for `x1` and for `x5`, the deviance differences for comparison with a straight line model (`m=2 vs lin.`) being respectively 19.3 and 31.1 at convergence (cycle 2). Predictors `x2`, `x3`, `x4b` and `x7` are dropped, as may be seen from their status `out` in the table `Final multivariable fractional polynomial model for rectime`.

Note that all predictors except `x4a` and `hormon` (which are binary) have been adjusted to the mean of the original variable. For example, the mean of `x1` (age) is 53.05 years. The first FP transformed variable for `x1` is `x1^-2` and is created by the expression `gen double Ix1__1 = X^-2-.0355 if e(sample)`. The value .0355 is obtained from (53.05/10)^−2. The division by 10 is applied automatically to improve the scaling of the regression coefficient for `Ix1__1`.

According to Sauerbrei and Royston (1999), medical knowledge dictates that the estimated risk function for `x5` (number of positive nodes), which was based on the above FP with powers $(-2,-1)$, should be monotonic, but it was not. They improved Model II by estimating a preliminary exponential transformation `x5e` $= \exp(-0.12*$`x5`$)$ for `x5` and fitting a degree 1 FP for `x5e`, thus obtaining a monotonic risk function. The value of -0.12 was estimated univariately using nonlinear Cox regression with the ado-file `boxtid` (Royston and Ambler 1999). To ensure a negative exponent Sauerbrei and Royston (1999) restricted the powers for `x5e` to be positive. Their Model III may be estimated using the following command:

```
. mfracpol cox rectime x1 x2 x3 x4a x4b x5e x6 x7 hormon, dead(censrec)
> alpha(.05) select(.05, hormon:1) df(x5e:2) xpowers(x5e:0.5 1 2 3)
```

Other than the customization for `x5e`, the command is the same as before. The resulting model is as reported in Table 4 of Sauerbrei and Royston (1999):

```
Final multivariable fractional polynomial model for rectime
------------------------------------------------------------

Variable |     -----Initial-----          -----Final-----
         |   df     Select   Alpha    Status    df    Powers
---------+--------------------------------------------------
      x1 |    4     0.0500   0.0500     in      4     -2 -.5
      x2 |    1     0.0500   0.0500     out     0
      x3 |    4     0.0500   0.0500     out     0
     x4a |    1     0.0500   0.0500     in      1     1
     x4b |    1     0.0500   0.0500     out     0
     x5e |    2     0.0500   0.0500     in      1     1
      x6 |    4     0.0500   0.0500     in      2     .5
      x7 |    4     0.0500   0.0500     out     0
  hormon |    1     1.0000   0.0500     in      1     1
---------+--------------------------------------------------

Cox regression -- Breslow method for ties
Entry time 0                                    Number of obs   =        686
                                                LR chi2(6)      =     153.11
                                                Prob > chi2     =     0.0000
Log likelihood = -1711.6186                     Pseudo R2       =     0.0428
------------------------------------------------------------------------------
 rectime |
 censrec |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
  Ix1__1 |   43.55382   8.253433      5.277   0.000       27.37738    59.73025
  Ix1__2 |  -17.48136   3.911882     -4.469   0.000      -25.14851   -9.814212
     x4a |   .5174351   .2493739      2.075   0.038       .0286713    1.006199
 Ix5e__1 |  -1.981213   .2268903     -8.732   0.000      -2.425909   -1.536516
  Ix6__1 |   -1.84008   .3508432     -5.245   0.000       -2.52772    -1.15244
  hormon |  -.3944998    .128097     -3.080   0.002      -.6455654   -.1434342
------------------------------------------------------------------------------
Deviance: 3423.237.
```

Technical note: Model selection procedures

Sauerbrei and Royston (1999)'s modifications to the algorithm of Royston and Altman (1994) were (a) to order the variables initially according to decreasing significance in a multiple linear regression model, and (b) to allow variables to have customized powers in special situations. As described above, Sauerbrei and Royston (1999) used the latter feature when modeling a variable which had been subjected to a preliminary transformation.

In what follows, we describe model selection procedures for a single continuous covariate x which represent one step of the iterative algorithm just exemplified. In each procedure, a significance level α_{sel} is chosen for testing for inclusion of x and another, α_{FP}, for comparisons between FP models. A variable x is forced into the model by setting $\alpha_{\text{sel}} = 1$. It is forced to assume the most complex functional form (i.e., highest degree FP) allowed for it by setting $\alpha_{\text{FP}} = 1$. Theoretically, any combination of α_{sel} and α_{FP} is possible, though in practice only a few choices are reasonable. For example, the choice $\alpha_{\text{sel}} = 1$, $\alpha_{\text{FP}} = 0.05$ (the default in `mfracpol`) includes x in the model and allows simplification of its functional form. The choice $\alpha_{\text{sel}} = \alpha_{\text{FP}} = 0.05$ additionally allows x to be dropped if it fails an overall test of significance at the 5% level. Full models may be built by taking $\alpha_{\text{sel}} = \alpha_{\text{FP}} = 1$. The combination $\alpha_{\text{sel}} = 0.05$, $\alpha_{\text{FP}} = 1$ is unlikely to be much used since x is either rejected or allowed full complexity, which seems rather perverse.

The null distribution of the likelihood-ratio statistic used in the significance tests is assumed to be F for normally distributed data, χ^2 in other cases. In the descriptions below, the most complex model allowed for x is taken to be an FP with $m = 2$, though the extension to $m > 2$ is obvious. Note that with the present update of `mfracpol` the complexity is not limited to $m = 2$; FP models with $m > 2$ are supported via the `df()` and `dfdefault()` options.

Previous procedure

In the earlier version of `mfracpol`, Royston and Ambler (1998) incorporated an initial variable inclusion step to reduce the Type I error rate. The procedure was as follows:

1. Perform a 4 df test at the α_{sel} level of the best-fitting second-degree FP against the null model. If the test is not significant, drop x and stop, otherwise continue.
2. Perform a 2 df test at the α_{FP} level of the best-fitting FP of degree 2 against the best FP of degree 1. If the test is significant, stop (the final model is the FP with $m = 2$), otherwise continue.
3. Perform a 1 df test at the α_{FP} level of the best-fitting FP of degree 1 against a straight line. The final model is the FP with $m = 1$ if the test is significant, otherwise it is a straight line.

When $\alpha_{\text{sel}} = 1$, step 1 is omitted. The main problem with this algorithm is that it can give illogical results. For example, it may happen that the inclusion test (step 1) is significant but that none of the subsequent tests ($m = 2$ vs $m = 1$, $m = 1$ vs straight line, or in fact straight line vs null, which is not formally part of the procedure) is significant. In this situation the procedure selects a straight line, which may even be the model least strongly supported by the data.

New default procedure

The model selection procedure described by Royston and Sauerbrei (1999) is implemented as the default in the present version of `mfracpol`. It has the flavor of a closed test (CT) procedure (Marcus et al. 1976) which maintains approximately the correct Type I error rate for each component test. The procedure allows the complexity of candidate models to increase progressively from a prespecified minimum—a null model if $\alpha_{\text{sel}} < 1$, or a straight line if $\alpha_{\text{sel}} = 1$—to a prespecified maximum—an FP—according to an ordered sequence of test results. The procedure is as follows:

1. Perform a 4 df test at the α_{sel} level of the best-fitting second-degree FP against the null model. If the test is not significant, drop x and stop, otherwise continue.
2. Perform a 3 df test at the α_{FP} level of the best-fitting second-degree FP against a straight line. If the test is not significant, stop (the final model is a straight line), otherwise continue.
3. Perform a 2 df test at the α_{FP} level of the best-fitting second-degree FP against the best-fitting first-degree FP. The final model is the FP with $m = 2$ if the test is significant, the FP with $m = 1$ if not.

The tests at steps 1, 2 and 3 are of overall association, nonlinearity and between a simpler or more complex FP model, respectively. When $\alpha_{\text{sel}} = 1$, step 1 is omitted.

The sequential procedure

For completeness and to facilitate further study, `mfracpol` with the `sequential` option performs Sauerbrei and Royston's (1999) version of Royston and Altman (1994)'s algorithm, which is as follows:

1. Perform a 2 df test at the α_{FP} level of the best-fitting FP of degree 2 against the best FP of degree 1. If the test is significant, stop (the final model is the FP with $m = 2$), otherwise continue.
2. Perform a 1 df test at the α_{FP} level of the best-fitting FP of degree 1 against a straight line. If the test is significant, stop (the final model is the FP with $m = 1$), otherwise continue.
3. Perform a 1 df test at the α_{sel} level of a straight line against the model omitting x. If the test is significant, the final model is a straight line, otherwise omit x.

   When $\alpha_{\text{sel}} = 1$, the final step is omitted.

Because several tests are carried out, when the true relationship is a straight line, the actual Type I error rate considerably exceeds the nominal value of α_{FP} (Ambler and Royston 1999). The procedure therefore tends to favor more complex models over simple ones and may be expected to overfit the data more than the new default procedure.

Acknowledgment

This work received financial support from project grant number 045512/Z/95/Z from the Wellcome Trust. We thank Dr. W. Sauerbrei for helpful comments on the manuscript.

References

Ambler, G. and P. Royston. 1999. Fractional polynomial model selection: some simulation results. *Journal of Statistical Computation and Simulation*, submitted.

Marcus, R., E. Peritz, and K. R. Gabriel. 1976. On closed test procedures with special reference to ordered analysis of variance. *Biometrika* 76: 655–660.

Royston, P. 1998. sg82: Fractional polynomials for st data. *Stata Technical Bulletin* 43: 32–32.

Royston, P. and D. G. Altman. 1994. Regression using fractional polynomials of continuous covariates: parsimonious parametric modelling (with discussion). *Applied Statistics* 43: 429–467.

Royston, P. and G. Ambler. 1998. sg81: Multivariable fractional polynomials. *Stata Technical Bulletin* 43: 24–32.

——. 1999. sg112: Nonlinear regression models involving power or exponential functions of covariates. *Stata Technical Bulletin* 49: 25–30.

Royston, P. and W. Sauerbrei. 1999. Test procedures for fractional polynomial model selection. In preparation for *Journal of the Royal Statistical Society*, Series A.

Sauerbrei, W. and P. Royston. 1999. Building multivariable prognostic and diagnostic models: transformation of the predictors using fractional polynomials. *Journal of the Royal Statistical Society*, Series A 162: 71–94.

sg97.1 | Revision of outreg

John Luke Gallup, Harvard University, john_gallup@harvard.edu

This revision of `outreg` adds enhancements and corrects a number of problems with the previous version in Gallup (1998). `outreg` has also been updated for Stata 6.0 and made more efficient because Stata has made fundamental changes in the way it reports estimation results (a great improvement).

`outreg` has several new capabilities. One can now:

- Choose different numbers of decimal places for each coefficient with the `bdec` option.
- Report the extra statistics appended to the `e(b)` matrix with the `xstats` option.
- Choose either single column or multiple column formatting for multi-equation estimation with the `onecol` option.
- Report p-values under coefficients with the `pvalue` option.
- Report the exponentiated form of coefficients in `logit`, `clogit`, `mlogit`, `glogit`, `cox`, `xtprobit`, `xtgee`, or any other command with the `eform` option.
- *varlists* can be specified for multivariate regressions.

Included with the current insert is a version of the `outreg` command written in Stata 5.0, `outreg5`, for backwards capability for those who have not upgraded yet, and for use with older routines that have not been updated for Stata 6.0 such as `dprobit2`, `dlogit2`, and `dmlogit2`. Given the major changes in the way Stata 6.0 reports results, it does not make sense to have a single `outreg` command that can work with both versions of Stata. There are probably still some Stata 5.0 estimation commands for which `outreg5` will not work correctly. Users of Stata 5.0 with the original `outreg` should switch to `outreg5` because it fixes a number of bugs in the original `outreg`.

As for those bugs, the most important was that the critical values used for determining asterisks to indicate significance levels were incorrect for nonlinear estimation (I didn't notice that `invt` is two-tailed, but `invnorm` is one-tailed). Also, despite my claims to the contrary, the original `outreg` did not work correctly with all Stata estimation commands. I have now tested `outreg` with all the commands. Please let me know if I have not tested thoroughly enough.

Reference

Gallup, J. L. 1998. sg97: Formatting regression output for published tables. *Stata Technical Bulletin* 46: 28.

sg107.1 | Generalized Lorenz curves and related graphs

Stephen P. Jenkins, ISER, University of Essex, UK, stephenj@essex.ac.uk
Philippe Van Kerm, GREBE, University of Namur, Belgium, philippe.vankerm@fundp.ac.be

A bug affecting the behavior of `glcurve` in some special cases has been found and fixed.

sg111	A modified likelihood-ratio test command

Santiago Perez-Hoyos, Institut Valencia d'Estudis en Salut Publica, Valencia, sperez@san.gva.es
Aurelio Tobias, Statistical Consultant, Madrid, bledatobias@ctv.es

Stata's `lrtest` command compares nested models estimated by maximum likelihood through the likelihood-ratio test (McCullagh and Nelder 1989) using a backward strategy; that is, to test if adding one or more variables improves the fit of the regression model. First the complete model containing all variables of interest must be estimated. The second model must be reduced and nested within the first, excluding those variables of interest. However, for nonstatisticians, a forward strategy seems more intuitive, that is, fitting the simplest model first and then testing the inclusion of the variable(s) of interest after that.

The `lrtest2` command presented in this insert is a simple modification of the original `lrtest` command, to perform the likelihood-ratio test under a forward strategy, although a backward strategy is also permitted.

Syntax

The `lrtest2` command has the same syntax and options as `lrtest`.

Example

Using the low birth weight data discussed in the Stata manual in the documentation for `lrtest`, we first fit a model adjusted by `race`, `smoke` and uterine irritability (`ui`).

```
. use birthwt
. describe

Contains data from birthwt.dta
  obs:           189
 vars:            10                          30 Jan 1998 12:59
 size:        15,876 (98.3% of memory free)
------------------------------------------------------------------------------
   1. low       double %9.0g                 birth weight less than 2.5kg (0
   2. age       double %9.0g                 age of mother in years
   3. lwt       double %9.0g                 weight of mother (lbs) last men
   4. race      double %9.0g                 white/black/other
   5. smoke     double %9.0g                 smoking status during the pregn
   6. ptl       double %9.0g                 number of previous premature la
   7. ht        double %9.0g                 history of hypertension (0/1)
   8. ui        double %9.0g                 has uterine irritability
   9. ftv       double %9.0g                 number of physician visits in f
  10. bwt       double %9.0g                 actual birth weight
------------------------------------------------------------------------------
Sorted by:

. xi: logistic low i.race smoke ui
i.race                Irace_1-3    (naturally coded; Irace_1 omitted)

Logit estimates                                   Number of obs   =        189
                                                  LR chi2(4)      =      18.80
                                                  Prob > chi2     =     0.0009
Log likelihood = -107.93404                       Pseudo R2       =     0.0801
------------------------------------------------------------------------------
     low | Odds Ratio   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
 Irace_2 |   3.052746   1.498084      2.274   0.023       1.166749    7.987368
 Irace_3 |   2.922593   1.189226      2.636   0.008        1.31646    6.488269
   smoke |   2.945742   1.101835      2.888   0.004        1.41517    6.131701
      ui |   2.419131   1.047358      2.040   0.041        1.03546    5.651783
------------------------------------------------------------------------------

. lrtest2, saving(0)
```

Now, we study the improvement of the goodness of fit of the logistic regression model including the variables `age`, weight at last menstrual period (`lwt`), premature labor history (`ptl`) and history of hypertension (`ht`).

```
. xi: logistic low age lwt i.race smoke ptl ht ui
i.race                Irace_1-3    (naturally coded; Irace_1 omitted)

Logit estimates                                   Number of obs   =        189
                                                  LR chi2(8)      =      33.25
                                                  Prob > chi2     =     0.0001
Log likelihood = -100.71348                       Pseudo R2       =     0.1417
------------------------------------------------------------------------------
```

```
     low | Odds Ratio   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+-------------------------------------------------------------------
     age |   .9732933   .0354791    -0.743   0.458       .9061816    1.045375
     lwt |   .9849321   .0068235    -2.192   0.028       .9716487    .9983972
  Irace_2 |   3.536789   1.862006     2.399   0.016       1.260315    9.925202
  Irace_3 |   2.367028   1.039593     1.962   0.050       1.000824    5.598207
   smoke |   2.517708   1.009244     2.303   0.021        1.14761    5.523529
     ptl |   1.719021   .5952396     1.565   0.118       .8720445    3.388628
      ht |   6.256967   4.328382     2.651   0.008       1.612606    24.27725
      ui |   2.135277   .9809289     1.651   0.099       .8677992    5.253991
-------------------------------------------------------------------------------

. lrtest2
Logistic:  likelihood-ratio test                      chi2(4)     =      14.44
                                                      Prob > chi2 =     0.0060
```

The result obtained is the same as using the `ltrest` command, following a backward strategy, concluding that the inclusion of the variables `age`, `lwt`, `ptl`, and `ht` improves the fit of the logistic regression model.

Reference

McCullagh D. W. and J. A. Nelder. 1989. *Generalized Linear Models*. London: Chapman and Hall.

sg112 | Nonlinear regression models involving power or exponential functions of covariates

Patrick Royston, Imperial College School of Medicine, UK, proyston@rpms.ac.uk
Gareth Ambler, Imperial College School of Medicine, UK, gambler@rpms.ac.uk

Introduction

A first degree fractional polynomial (FP) is a function of the form $\beta_0 + \beta_1 x^p$, where $x > 0$ and p is a power chosen from the set $\mathcal{P} = \{-2, -1, -0.5, 0, 0.5, 1, 2, 3\}$. By convention, x^p with $p = 0$ means $\log x$. The best-fitting (maximum likelihood) value $\widetilde{p}$ of p in $\mathcal{P}$ may be found by using the Stata command `fracpoly` with the option `degree(1)`. A first degree FP is a special case of the power-function family (Box and Tidwell 1962) which is obtained when p is allowed to take any real value, i.e., is not restricted to $\mathcal{P}$. Power functions yield curves for $p \neq 1$ and straight lines for $p = 1$. For $p < 0$ they have an asymptote β_0 as $x \to \infty$. The main purpose of the present insert is to describe a program `boxtid` which finds the maximum likelihood estimate $\widehat{p}$ of p for several types of error structure, most importantly normal and binomial errors, GLMs and Cox regression. Closely related is the family of exponential functions $\beta_0 + \beta_1 \exp(px)$ which, since $\exp(px) = [\exp(x)]^p$, may be viewed as power functions on the exponential scale. `boxtid` also estimates these models. In addition, multivariable nonlinear models with power or exponential transformations of several x's may be estimated.

There are two main reasons why it may be useful to fit a Box–Tidwell model rather than or in addition to a first degree FP. First, the fit may be markedly better when $\widehat{p}$ lies considerably outside the interval $[-2, 3]$ or lies between elements of $\mathcal{P}$. If $\widehat{p}$ is close to $\widetilde{p}$ (say, within one standard error of it), we are reassured that the FP model has not missed anything important. Since `boxtid` can fit continuous powers for any degree of FP, it can be used to check the appropriateness of powers for any such FP. Second, `boxtid` can estimate confidence intervals for p and for the fitted values $\widehat{\beta}_0 + \widehat{\beta}_1 x^{\widehat{p}}$ which allow for the estimation of p. Confidence intervals for fitted values are really only achievable with FPs by the use of bootstrapping, which is computationally intensive and not straightforward to set up.

To demonstrate the method we give examples using the well-known `auto` dataset supplied with Stata, a breast cancer dataset previously analysed by Sauerbrei and Royston (1999), an IgG dataset previously analysed by Royston and Altman (1994) and a dataset of measurements of fetal growth.

The ado-file `boxtid` is a regression-like command with the following basic syntax:

`boxtid` *regression_cmd yvar xvarlist* [*weight*] [`if` *exp*] [`in` *range*] [`,` *options*]

Details are given in the section *Syntax*.

Example 1: Automobile data

We use `boxtid` to fit a Box–Tidwell model to predict `mpg` from `weight` for the dataset `auto.dta` as follows:

```
. boxtid regress mpg weight
Iteration 0:  Deviance =  385.8874
Iteration 1:  Deviance =  385.8874 (change = -.0000794)
```

```
-> gen double Iweig__1 = X^-0.4460-.6109 if e(sample)
-> gen double Iweig__2 = X^-0.4460*ln(X)-.6751 if e(sample)
   (where: X = weight/1000)
[Total iterations: 1]
Box-Tidwell regression model
  Source |       SS       df       MS                  Number of obs =      74
---------+------------------------------               F(  2,    71) =   73.33
   Model |  1646.43761     2  823.218806               Prob > F      =  0.0000
Residual |  797.021847    71  11.2256598               R-squared     =  0.6738
---------+------------------------------               Adj R-squared =  0.6646
   Total |  2443.45946    73  33.4720474               Root MSE      =  3.3505
------------------------------------------------------------------------------
     mpg |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
Iweig__1 |    63.2036   52.77866      1.198   0.235      -42.03406    168.4413
Iweig_p1 |     -.0087   42.75559      0.000   1.000      -85.26092    85.24352
   _cons |   20.42983   .5515679     37.040   0.000       19.33004    21.52963
------------------------------------------------------------------------------
weight   |  -.0060087   .0005179    -11.603   Nonlin. dev. 4.890   (P = 0.031)
      p1 |  -.4459573   .6584836     -0.677
------------------------------------------------------------------------------
Deviance:  385.887.
```

The estimation procedure converges after one iteration, showing that the initial value of p was very accurate. The above table shows that the maximum likelihood estimate $\widehat{p} = -0.446$ (SE 0.658). The deviance of the model is 385.887. A first degree FP has $\widetilde{p} = -0.5$ and a deviance of 385.894, so the two models are essentially identical in this case. The entry marked `Nonlin. dev.` indicates the amount of nonlinearity found in the data. In this case the deviance difference between the Box–Tidwell model and a straight line is 4.89 ($P = 0.031$), so there is mildly significant nonlinearity, at least in p-value terms.

Two new variables are created for each power estimated. The first (`Iweig__1` above, with power `p1`) is the transformed predictor variable. The second (`Iweig_p1` above) is an auxiliary variable used within the algorithm to estimate p. (In the above output, `Iweig_p1` is initially called `Iweig__2` but is immediately renamed.) At convergence the auxiliary variable has (or should have) a coefficient estimate close to zero. Its presence in the final regression model ensures that the standard errors are valid. Without it, the fitted values from the model would be the same but the standard errors would be seriously underestimated.

Figure 1 shows the observed and fitted values of `mpg` from the model, together with their standard errors.

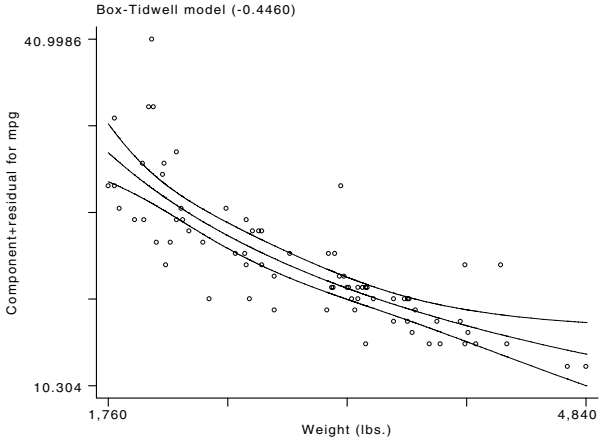

Figure 1: Observed and fitted values of `mpg` and their standard errors from a Box–Tidwell model

The plot was obtained simply by typing `fracplot` (`boxtid` is fully compatible with the `fracplot` and `fracpred` commands in Stata 6). The nonlinearity in the relationship between `mpg` and `weight` can be clearly seen.

Figure 2 shows the standard errors of the predicted values of `mpg` from the FP and Box–Tidwell models plotted against `weight`.

(*Graph on next page*)

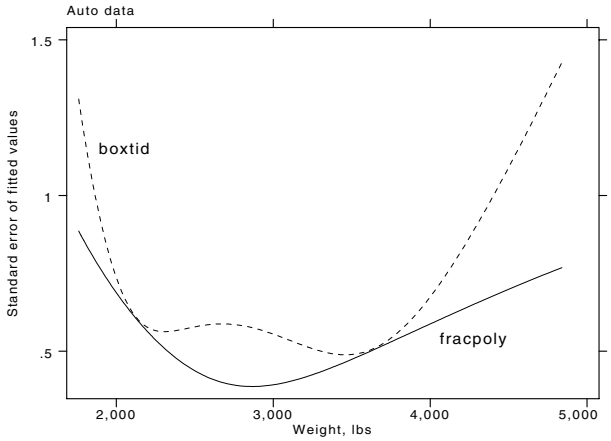

Figure 2: SEs of predicted values of `mpg` from FP (solid line) and Box–Tidwell (dashed line) models.

The SEs were obtained by using `fracpred` with the option `stdp` immediately after running `fracpoly` and `boxtid`. Except at car weights of 2200 and 3700 lbs, the standard errors from `boxtid` are markedly larger than from `fracpoly`, showing that the estimation of p as a continuous parameter may make a major difference.

Example 2: Recurrence-free survival time in breast cancer

The dataset consists of information on 686 patients with primary node positive breast cancer who were recruited by the German Breast Cancer Study Group (GBSG) between July 1984 and December 1989. Of these, 299 patients experienced at least one disease recurrence or died during the follow-up period. The median follow-up time was nearly 5 years. The data have been extensively analysed by Sauerbrei and Royston (1999), who used fractional polynomials to develop prognostic models. Here we consider Cox regression models for the relationship between recurrence-free survival time (`rectime`) with censoring variable `censrec` and the strongest prognostic factor, the number of positive lymph nodes (`x5`).

The best-fitting second degree FP has powers $(1, 2)$ and a deviance of 3494.99. The model fits significantly better than first degree FP and Box–Tidwell models. However, the second degree FP model is a quadratic curve with a maximum log relative hazard estimated at 24 positive nodes. Such a maximum implies that the risk of disease recurrence actually decreases for patients with >24 nodes, which is strongly contrary to medical knowledge. To produce a risk curve consistent with medical knowledge, Sauerbrei and Royston (1999) fitted a univariate exponential model to obtain a preliminary transformation $\texttt{x5e} = \exp(p * \texttt{x5})$. They then used the ado-file `mfracpol` (Royston and Ambler 1998, 1999) to model `x5e` simultaneously with other prognostic factors in a multivariable FP model.

The exponential model may be fit by `boxtid` using the command

```
. boxtid cox rectime x5, dead(censrec) expon(x5)
```

The estimate $\widehat{p} = -0.117$ (SE 0.042). The fitted curve from this model is monotonic and has an asymptote. The deviance is 3.0 higher than that of the second degree FP. We illustrate the different fits in Figure 3.

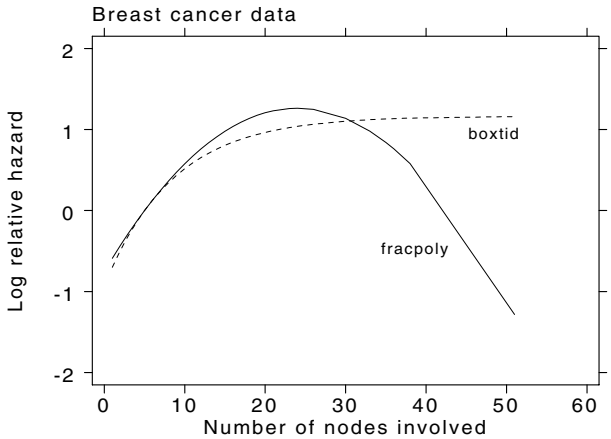

Figure 3: Fitted log relative hazard functions from FP and Box–Tidwell (exponential) models

Example 3: Checking FP models

Breast cancer data (continued)

Sauerbrei and Royston's (1999) final model (III) may be checked using `boxtid`. As well as `x5e`, model III includes the predictors age (`x1`), tumor grade (`x4a`), progesterone receptor status (`x6`) and hormonal treatment status (`hormon`). FPs were used for `x1` $(-2\ -0.5)$ and `x6` (0.5), while `x4a`, `x5e` and `hormon` were entered as linear. We check the `x5` and `x6` transformations simultaneously by fitting a multivariable power and exponential model. The other predictors are entered in the model as linear terms. We fit this model using the commands

```
. fracgen x1 -2 -0.5
. boxtid cox rectime x1_1 x1_2 x4a x5 x6 hormon, dead(cens) expon(x5) df(1, x5 x6:2)
```

The value of $\widehat{p}$ for `x6` is 0.256 (SE 0.181). The fitted curve for `x6` is similar to that of the best-fitting first degree FP curve.

Fetal femur length data

Measurements of the femur length of 649 fetuses were obtained by ultrasound scanning of the mother's abdomen. A log transformation of femur length removes almost all of the heteroscedasticity seen in the untransformed observations. Figure 4 shows a scatter plot of log(femur length) against gestational age, with the fitted curves from a second degree FP inscribed.

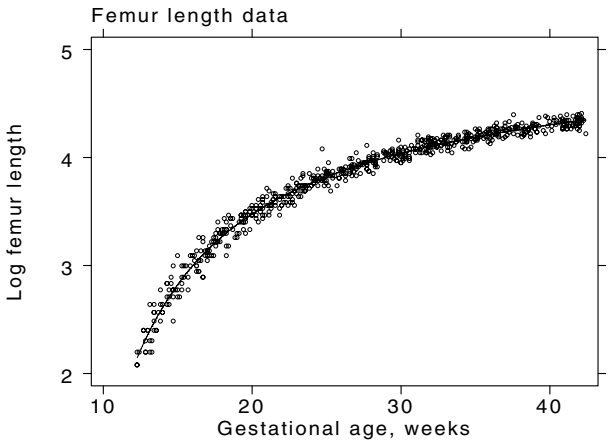

Figure 4: Log transformed femur length data with fitted second degree FP

First and second degree FPs have powers $\widetilde{p}$ of -1 and $(-2, 0)$ and deviances of -1540.02 and -1689.93 respectively. The large deviance difference of 149.91 shows that a second degree FP is a much better fit than a first degree. However, a first degree Box–Tidwell model has a deviance of -1683.99, close to that of the *second* degree FP. The power $\widehat{p} = -1.39$ has SE 0.03, so here the power is very precisely estimated and is some 13 standard errors away from the first degree FP power of -1. Moreover the Box–Tidwell model is monotonic, which is appropriate for growth data since average femur length does not diminish as gestation advances. Monotonicity is not guaranteed with second degree FP models. The fitted curves from the second degree FP and first degree Box–Tidwell models are almost superimposable. In this case we conclude that a first degree Box–Tidwell model is probably preferable to a first or second degree FP model.

Immunoglobulin-G (IgG) data

The IgG data were used as an example for second degree FPs (see [R] **fracpoly**, pp. 502–504). A measurement of IgG was made on each of 298 children aged 6 months to 6 years. The outcome variable is the square root of the IgG concentration. The best-fitting second degree FP has powers $(-2, 2)$, so the fitted model is of the form $b_0 + b_1 x^{-2} + b_2 x^2$. The best-fitting second degree Box–Tidwell model has powers $(-2.58, 1.85)$ with very large SEs of $(2.21, 1.44)$. The deviance difference between the FP and Box–Tidwell models is small (0.18) and the fits are almost identical. In this case we are reassured that the FP model cannot be improved by a Box–Tidwell model.

In our experience with real datasets, second degree FP models provide good coverage of the two dimensional power space and second degree Box–Tidwell models are seldom an improvement.

Syntax

`boxtid` *regression_cmd yvar xvarlist* [*weight*] [`if` *exp*] [`in` *range*] [`,` *major_options minor_options*

regression_cmd_options]

where *regression_cmd* may be one of `cox`, `glm`, `logistic`, `logit`, `poisson`, `regress`; `boxtid` shares the features of all estimation commands; `fracplot` may be used following `boxtid` to show plots of fitted values and partial residuals; `fracpred` may be used for prediction; and all weight types supported by *regression_cmd* are allowed.

Options

The *major_options* (most used options) are

`adjust(`*adj_list*`)` `df(`*df_list*`)` `expon(`*varlist*`)`

and the *minor_options* are

`dfdefault(#)` `init(`*init_list*`)` `iter(#)` `ltolerance(#)` `powers(`*numlist*`)` `trace` `zero(`*varlist*`)`

regression_cmd_options are any of the options available with *regression_cmd*.

Major options

`adjust(`*adj_list*`)` defines the adjustment for the covariates *xvar1, xvar2, . . . , xvarlist*. The default is `adjust(mean)`, except for binary covariates where it is `adjust(#)`, # being the lower of the two distinct values of the covariate. A typical item in *adj_list* is *varlist*`: mean|#|no`. Items are separated by commas. The first item is special in that *varlist*`:` is optional, and if omitted, the default is (re)set to the specified value (`mean` or # or `no`). For example, `adjust(no, age:mean)` sets the default to `no` and adjustment for `age` to `mean`.

`df(`*df_list*`)` sets up the degrees of freedom (df) for each predictor. Each power and each regression coefficient count as 1 df. Predictors specified to have 1 df are fitted as linear terms in the model. The first item in *df_list* may be either # or *varlist* `:#`. Subsequent items must be *varlist*`:#`. Items are separated by commas and *varlist* is specified in the usual way for variables. With the first type of item, the df for all predictors are taken to be #. With the second type of item, all members of *varlist* (which must be a subset of *xvarlist*) have # df.

The default df for a predictor (specified in *xvarlist* but not in *df_list*) are assigned according to the number of distinct (unique) values of the predictor as follows:

No. of distinct values	**Default df**
1	(not applicable)
2–3	1
4–5	min(2, `dfdefault(#)`)
$\geq$6	`dfdefault(#)`

`expon(`*varlist*`)` specifies that all members of *varlist* are to be modeled using exponential functions, the default being power (Box–Tidwell) functions. For each *xvar* in *varlist*, a multi-exponential model

$$\beta_1 \exp(p_1 x) + \beta_2 \exp(p_2 x) + \ldots$$

is estimated.

Minor options

`dfdefault(#)` determines the default maximum degrees of freedom (df) for a predictor. The default is 2.

`init(`*init_list*`)` sets initial values for the power parameters of the model. By default these are calculated automatically. The first item in *init_list* may be either # [# . . .] or *varlist*:# [# . . .]. Subsequent items must be varlist:# [# . . .]. Items are separated by commas and *varlist* is specified in the usual way for variables. If the first item is # [# . . .], this becomes the default initial value for all variables, but subsequent items (re)set the initial value for variables in subsequent *varlist*s. If the df for a variable in the model is $d > 1$ then # # . . . consists of $d/2$ items. Typically $d = 2$ so that there is just one initial value, #.

`iter(#)` sets # to be the maximum number of iterations allowed for the fitting algorithm to converge. The default is 100.

`ltolerance(#)` is the maximum difference in deviance between iterations required for convergence of the fitting algorithm. The default is 0.001.

`powers(`*numlist*`)` defines the powers to be used with fractional polynomial initialization for *xvarlist*.

`trace` reports the progress of the fitting procedure towards convergence.

`zero(`*varlist*`)` transforms negative and zero values of all members of *varlist* (a subset of *xvarlist*) to zero before fitting the model.

Fitted values, standard errors, graphs

Fitted values and standard errors from a Box–Tidwell model may be obtained by using Stata's `fracpred` command. The fitted functions for each predictor may be plotted using Stata's `fracplot` command.

Note

Please ensure that you have a release or update of Stata 6 no earlier than 4 March 1999. The update of 4 March 1999 contains some important changes to `fracpoly` which affect `boxtid`.

Acknowledgment

The research received financial support from project grant number 045512/Z/95/Z from The Wellcome Trust.

References

Box, P. W. and P. W. Tidwell. 1962. Transformation of the independent variables. *Technometrics* 4: 531–550.

Royston, P. and D. G. Altman. 1994. Regression using fractional polynomials of continuous covariates: parsimonious parametric modelling (with discussion). *Applied Statistics* 43: 429–467.

Royston, P. and G. Ambler. 1998. sg81: Multivariable fractional polynomials. *Stata Technical Bulletin* 43: 24–32.

——. 1999. sg81.1: Multivariable fractional polynomials: update. *Stata Technical Bulletin* 49: 17–23.

Sauerbrei, W. and P. Royston. 1999. Building multivariable prognostic and diagnostic models: transformation of the predictors using fractional polynomials. *Journal of the Royal Statistical Society*, Series A 162: 71–94.

ssa13 Analysis of multiple failure-time data with Stata

Mario Cleves, Stata Corporation, mcleves@stata.com

1. Introduction

Multiple failure-time data or multivariate survival data are frequently encountered in biomedical and other investigations. These data arise from time-to-occurrence studies when either of two or more events (failures) occur for the same subject, or from identical events occurring to related subjects such as family members or classmates. In these studies, failure times are correlated within cluster (subject or group), violating the independence of failure times assumption required in traditional survival analysis.

In this paper we follow Therneau's (1997) suggestion that for analyses purposes, failure events be classified according to (1) whether they have a natural order, and (2) whether they are recurrences of the same types of events. Failures of the same type include, for example, repeated lung infections with pseudomonas in children with cystic fibrosis, or the development of breast cancer in genetically predisposed families. Failures of different types include adverse reactions to therapy in cancer patients on a particular treatment protocol, or the development of connective tissue disease symptoms in a group of third graders exposed to hazardous waste.

Ordered events may result from a study that records the time to first myocardial infarction (MI), second MI, and so on. These are ordered events in the sense that the second event can not occur before the first event. Unordered events, on the other hand, can occur in any sequence. For example, in a study of liver disease patients, a panel of 7 liver function laboratory tests can become abnormal in a specific order for one patient and in different order for another patient. The order in which the tests become abnormal (fail) is random.

The simplest way of analyzing multiple failure data is to examine time to first event, ignoring additional failures. This approach, however, is usually not adequate because it wastes possibly relevant information. Alternative methods have been developed that make use of all available data while accounting for the lack of independence of the failure times. Two approaches to modeling these data have gained popularity over the last few years. In the first approach, the frailty model method, the association between failure times is explicitly modeled as a random-effect term, called the frailty. Frailties are unobserved effects shared by all members of the cluster. These unmeasured effects are assumed to follow a known statistical distribution, often the gamma distribution, with mean equal to one and unknown variance. This paper will not consider frailty models further.

In the second approach, the dependencies between failure times are not included in the models. Instead, the covariance matrix of the estimators is adjusted to account for the additional correlation. These models, which we will call "variance-corrected"

models, are easily estimated in Stata. In this paper we illustrate the principal ideas and procedures for estimating these models using the Cox proportional hazard model. There is no theoretical reason, however, why other hazard functions could not be used.

2. Methods

Let X_{ki} and C_{ki} be the failure and censoring time of the k^{th} failure type $(k = 1, ..., K)$ in the i^{th} cluster $(i = 1, ..., m)$, and let Z_{ki} be a $p-$vector of possibly time-dependent covariates, for i^{th} cluster with respect to the k^{th} failure type. "Failure type" is used here to mean both failures of different types, and failures of the same type. Assume that X_{ki} and C_{ki} are independent, conditional on the covariate vector (Z_{ki}). Define $T_{ki} = \min(X_{ki}, C_{ki})$ and $\delta_{ki} = I(X_{ki} \leq C_{ki})$ where $I(.)$ is the indicator function, and let β be a $p-$vector of unknown regression coefficients. Under the proportional hazard assumption, the hazard function of the i^{th} cluster for the k^{th} failure type is

$$\lambda_k(t; Z_{ki}) = \lambda_0(t)e^{Z_{ki}\beta} \tag{1}$$

if the baseline hazard function is assumed to be equal for every failure type, or

$$\lambda_k(t; Z_{ki}) = \lambda_{0k}(t)e^{Z_{ki}\beta} \tag{2}$$

if the baseline hazard function is allowed to differ by failure type (Lin 1994).

Maximum likelihood estimates of β for models (1) or (2) are obtained from the Cox's partial likelihood function, $L(\beta)$, assuming independence of failure times. The estimator $\hat{\beta}$ has been shown to be a consistent estimator for β and asymptotically normal as long as the marginal models are correctly specified (Lin 1994). The resulting estimated covariance matrix obtained as the inverse of the information matrix, however,

$$I^{-1} = -\partial^2 \log L(\beta)/\partial\beta\partial\beta'$$

does not take into account the additional correlation in the data, and therefore, it is not appropriate for testing or constructing confidence intervals for multiple failure time data.

Lin and Wei (1989) proposed a modification to this naive estimate, appropriate when the Cox model is misspecified. The resulting robust variance–covariance matrix is estimated as

$$V = I^{-1}U'UI^{-1}$$

where U is a n x p matrix of efficient score residuals. The above formula assumes that the n observations are independent. When observations are not independent, but can be divided into m independent groups $(G_1, G_2, ..., G_m)$, then the robust covariance matrix takes the form

$$V = I^{-1}G'GI^{-1}$$

where G is a m x p matrix of the group efficient score residuals. In terms of Stata, V is calculated according to the first formula when the `robust` option is specified and according to the second formula when `cluster()` is also specified. (`cluster()` implies `robust` in Stata, so specifying `cluster()` by itself is adequate).

3. Implementation and examples

All variance-adjusted models suggested to date can be estimated in Stata. All that is required is some preliminary thought about the analytic model required, the correct way to set up the data, and the command options to be specified.

The examples in this section are presented under the following headings

Unordered failure events
  Unordered failure events of the same type
  Unordered failure events of different types (competing risk)
Ordered failure events
  The Andersen–Gill model
  The marginal risk set model
  The conditional risk set model (time from entry)
  The conditional risk set model (time from the previous event)

All the examples we will described use the survival time (`st`) system, which is to say, for instance, in terms of `stcox` rather than `cox`. Although it is not necessary that the `st` system be used, it is recommended.

The steps for analyzing multiple failure data in Stata are (1) decide whether the failure events are ordered or unordered, (2) select the proper statistical model for the data, (3) organize the data according to the model selected, and (4) use the proper commands and command options to `stset` the data and estimate the model. Much of this paper deals with the appropriate method for setting the data and the correct way of specifying the estimation command. The examples are used solely to illustrate these processes. Consult the references for more detail discussions on these methods and the datasets used.

3.1 *Unordered failure events*

The data setup for the analysis of unordered events is relatively simple. One first decides if the failure events are of the same type or of different type, or equivalently, whether the baseline hazard should be equal for all event types or should be allowed to vary by event type. Failure events of the same type are described in section 3.1.1. In section 3.1.2, the baseline hazard is allowed to vary by failure type and is used to examine a competing-risk dataset.

3.1.1 *Unordered failure events of the same type*

A possible source of correlated failure times of the same event type are familial studies, in which each family member is at risk of developing a disease of interest. Failure times of family members are correlated because they share genetic and perhaps environmental factors.

Another source of correlated failure times of the same type are studies where the same event can occur on the same individual multiple times. This is rare because we are also restricting the events to have no order. Lee, Wei, and Amato (1992) analyzed data from the National Eye Institute study on the efficacy of photocoagulation as a treatment for diabetic retinopathy. In that study, each subject was treated with photocoagulation on one randomly selected eye while the other eye served as an untreated matched control. The outcome of interest was the onset of severe visual loss, and the study hoped to show that laser photocoagulation significantly reduced the time to onset of blindness. In this study, the sampling units, the eyes, are pairwise correlated, the failure types are the same, and unordered because the right eye can fail before the left eye or vice versa.

These types of data are straightforward to setup and analyze in Stata. Each sampling unit is entered once into the dataset. In the family data, each family member appears as an observation in the dataset and an id variable identifies his or her family. In the laser photocoagulation example, because each eye is a sampling unit, each eye appears as an observation in the dataset. Therefore, if there are n patients in the diabetic retinopathy study then the resulting dataset would contain $2n$ observations. A variable is used to identify the matched eyes.

We will illustrate using a subset of the diabetic retinopathy data. The data from 197 high-risk patients was entered into a Stata dataset. The first four observations are

```
. list in 1/4, noobs
   id       time       cens     agegrp      treat
    5      46.23          0          1          1
    5      46.23          0          1          0
   14       42.5          0          0          1
   14       31.3          1          0          0
```

Each patient has two observations in the dataset, one for the treated eye (`treat==1`) and another for the "control" eye, `treat==0`. The data, therefore, contain 394 observations. Each eye is assumed to enter the study at time 0 and it is followed until blindness develops or censoring occurs. The follow-up time is given by the variable `time`. The four observations listed above correspond to patients with `id`=5 and `id`=14.

After creating the dataset, it is then `stset` as usual, however the `id()` option is not specified. Specifying `id()` would cause `stset` to interpret subjects with the same `id()` as the same sampling unit and would drop them because of overlapping study times. Thus, we type

```
. stset time, failure(cens)
     failure event:  cens ~= 0 & cens ~= .
obs. time interval:  (0, time]
 exit on or before:  failure
--------------------------------------------------------------------------
      394  total obs.
        0  exclusions
--------------------------------------------------------------------------
      394  obs. remaining, representing
      155  failures in single record/single failure data
 14018.24  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =     74.97
```

Note that `stset` correctly reports that there are 394 observations. The command for estimating the corresponding Cox model is

```
. stcox agegrp treat, cluster(id) efron nohr
Iteration 0:   log likelihood = -867.98581
Iteration 1:   log likelihood = -856.74901
Iteration 2:   log likelihood = -856.74456
Refining estimates:
Iteration 0:   log likelihood = -856.74456
Cox regression -- Efron method for ties
No. of subjects =          394                     Number of obs   =       394
No. of failures =          155
Time at risk    =  14018.24001
                                                   Wald chi2(2)    =     27.71
Log likelihood  =   -856.74456                     Prob > chi2     =    0.0000
                              (standard errors adjusted for clustering on id)
------------------------------------------------------------------------------
      _t |               Robust
      _d |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
  agegrp |   .0538829   .1790951      0.301   0.764      -.2971371    .4049028
   treat |  -.7789297   .1488857     -5.232   0.000       -1.07074    -.487119
------------------------------------------------------------------------------
```

The `cluster(id)` option specifies to `stcox` which observations are related. Stata knows to produce robust standard errors whenever the `cluster()` option is used. The `efron` option requests that Efron's method for handling ties be used and the `nohr` option is used to request that coefficients, instead of hazard ratios, be reported.

3.1.2 *Unordered failure events of different types (competing risk)*

A common data source of unordered failure events of different types are competing-risk studies. In these studies, a patient can suffer several outcomes of interest in random order. In the analysis of these data, the baseline hazard function is allowed to vary by failure type. This is accomplished by stratifying the data on failure type, allowing each stratum to have its own baseline hazard function, but restricting the coefficients to be the same across strata.

We illustrate the use of Stata in the analysis of a competing-risk model, with a subset of the Mayo Clinic's Ursodeoxycholic acid (UDCA) data (Lindor et al. 1994). The data consists of 170 patients with primary biliary cirrhosis randomly allocated to either the UDCA treatment group or a group receiving a placebo. The times up to nine possible events were recorded: death, liver transplant, voluntary withdraw, histologic progression, development of varices, development of ascites, development of encephalophathy, doubling of bilirubin, and worsening of symptoms. All times were measured from the date of treatment allocation.

An important characteristic of these failure events is that each can occur only once per subject. Note that all subjects are at risk for all events, and also, that when a subject experiences one of the events, he remains at risk for all other events. Therefore, if there are k possible events, each subject will appear k times in the dataset, once for each possible failure. Here is the resulting data for two of the subjects.

```
. list id rx bili time status rec if id==5 | id==18,nod noobs
      id         rx       bili       time     status        rec
       5    placebo   .0953102       1875          0          1
       5    placebo   .0953102       1875          0          2
       5    placebo   .0953102       1875          0          3
       5    placebo   .0953102       1875          0          4
       5    placebo   .0953102       1875          0          5
       5    placebo   .0953102       1875          0          6
```

```
  5    placebo   .0953102       1875          0          7
  5    placebo   .0953102       1875          0          8
  5    placebo   .0953102       1875          0          9
 18    placebo   .1823216        391          1          9
 18    placebo   .1823216        391          1          8
 18    placebo   .1823216        763          1          5
 18    placebo   .1823216        765          0          2
 18    placebo   .1823216        765          0          1
 18    placebo   .1823216        765          0          6
 18    placebo   .1823216        765          0          7
 18    placebo   .1823216        765          1          3
 18    placebo   .1823216        765          0          4
```

Each patient appears nine times, once for each possible event. The event type, `rec`, is coded as 1 through 9. Patient number 5, did not experience any events during the 1,875 days of follow-up. Thus, he appears censored nine times in the data, each observation recording the complete follow-up period. Patient 18 experienced 4 events: `rec`=8 (doubling of bilirubin), `rec`=9 (worsening of symptoms), `rec`=5 (development of varices) and `rec`=3 (voluntary withdraw).

The command to `stset` the data is used without specifying the `id()` option.

```
. stset time, failure(status)
     failure event:  status ~= 0 & status ~= .
obs. time interval:  (0, time]
 exit on or before:  failure
------------------------------------------------------------------------------
     1530  total obs.
        0  exclusions
------------------------------------------------------------------------------
     1530  obs. remaining, representing
      145  failures in single record/single failure data
  1808720  total analysis time at risk, at risk from t =         0
                            earliest observed entry t =         0
                                 last observed exit t =      1896
```

It correctly reported 1,530 observations (170x9).

The `id` variable will be used to cluster the related observations when estimating the Cox model. Additionally, it does not seem reasonable to assume that each failure type should have the same baseline hazard, thus the Cox model will be stratified by failure type.

```
. stcox rx bili hi_sta, nohr efron robust strata(rec) cluster(id) nolog
Stratified Cox regr. -- Efron method for ties
No. of subjects =         1530                     Number of obs   =      1530
No. of failures =          145
Time at risk    =      1808720
                                                   Wald chi2(3)    =     31.99
Log likelihood  =   -662.44704                     Prob > chi2     =    0.0000
                              (standard errors adjusted for clustering on id)
------------------------------------------------------------------------------
      _t |               Robust
      _d |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
      rx |  -.9371209    .240996     -3.889   0.000      -1.409464   -.4647774
    bili |   .5859002   .1491832      3.927   0.000       .2935065    .8782939
hi_stage |  -.0754988   .2777845     -0.272   0.786      -.6199464    .4689488
------------------------------------------------------------------------------
                                                          Stratified by rec
```

The covariates are treatment group (`rx`), log(bilirubin)(`bili`), and high histologic stage indicator (`hi_stage`).

3.2 *Ordered failure events*

There are several approaches to the analysis of ordered events. The principal difference between these methods is in the way that the risk sets are defined at each failure time. The simplest method to implement in Stata follows the counting process approach of Andersen and Gill (1982). The basic assumption is that all failure types are equal or indistinguishable. The problem then reduces to the analysis of time to first event, time to second event, and so on. Thus, the risk set at time t for event k, is all subjects under observation at time t. A major limitation of this approach is that it does not allow more than one event to occur at a given time. For example, in a study examining time to side effects of a new medication, if a patient exhibits two

side effects at the same time, the corresponding observations are dropped because the time span between failures is zero. This approach is illustrated in section 3.2.1.

A second model, proposed by Wei, Lin, and Weissfeld (1989), is based on the idea of marginal risk sets. For this analysis, the data is treated like a competing risk dataset, as if the failure events were unordered, so each event has its own stratum and each patient appears in all strata. The marginal risk set at time t for event k, is made up of all subjects under observation at time t that have not had event k. This approach is illustrated in section 3.2.2.

A third method proposed by Prentice, Williams, and Peterson (1981) is known as the conditional risk set model. The data is setup as for Andersen and Gill's counting processes method, except that the analysis is stratified by failure order. The assumption made is that a subject is not at risk of a second event until the first event has occurred and so on. Thus the conditional risk set at time t for event k, is made up of all subjects under observation at time t, that have had event $k-1$. There are two variations to this approach. In the first variation, time to each event is measured from entry time, and in the second variation, time to each event is measured from the previous event. This approach is illustrated in sections 3.2.3 and 3.2.4.

The above three approaches will be illustrated using the bladder cancer data presented by Wei, Lin, and Weissfeld (1989). These data were collected from a study of 85 subjects randomly assigned to either a treatment group receiving the drug thiotepa or to a group receiving a placebo control. For each patient, time for up to four tumor recurrences was recorded in months (`r1` – `r4`). These are the first nine observations in the data.

```
. list in 1/9,noobs nod
    id     group  futime  number    size      r1      r2      r3      r4
     1   placebo       1       1       3       0       0       0       0
     2   placebo       4       2       1       0       0       0       0
     3   placebo       7       1       1       0       0       0       0
     4   placebo      10       5       1       0       0       0       0
     5   placebo      10       4       1       6       0       0       0
     6   placebo      14       1       1       0       0       0       0
     7   placebo      18       1       1       0       0       0       0
     8   placebo      18       1       3       5       0       0       0
     9   placebo      18       1       1      12      16       0       0
```

The `id` variable identifies the patients, `group` is the treatment group, `futime` is the total follow-up time for the patient, `number` is the number of initial tumors, `size` is the initial tumor size, and `r1` to `r4` are the times to first, second, third, and fourth recurrence of tumors. A recurrence time of zero indicates no tumor.

3.2.1 *The Andersen–Gill model*

To implement the Andersen and Gill model using the results from the bladder cancer study, the data are set up as follows: for each patient there must be one observation per event or time interval. For example, if a subject has one event, then there will be two observations for that subject. The first observation will cover the time span from entry into the study until the time of the event, and the second observation spans the time from the event to the end of follow-up. The data for the nine subjects listed above is

```
. list if id<10, noobs nod
    id     group       time0    time  status  number    size
     1   placebo           0       1       0       1       3
     2   placebo           0       4       0       2       1
     3   placebo           0       7       0       1       1
     4   placebo           0      10       0       5       1
     5   placebo           0       6       1       4       1
     5   placebo           6      10       0       4       1
     6   placebo           0      14       0       1       1
     7   placebo           0      18       0       1       1
     8   placebo           0       5       1       1       3
     8   placebo           5      18       0       1       3
     9   placebo           0      12       1       1       1
     9   placebo          12      16       1       1       1
     9   placebo          16      18       0       1       1
```

In the original data, subjects 1 through 4 had no tumors recur, thus, each of these 4 patients has only one censored (`status`=0) observation spanning from `time0`=0 to end of follow-up (`time`=`futime`). Patient 5 (`id`=5) had one tumor recur at 6 months and was followed until month 14. This patient has two observations in the final dataset; one from `time0`=0 to tumor recurrence (`time`=6), ending in an event (`status`=1), and another from `time0`=6 to end of follow-up (`time`=10), ending as censored (`status` =0).

We `stset` the data with the command

```
. stset time,fail(status) exit(futime) id(id) enter(time0)
               id:  id
     failure event:  status ~= 0 & status ~= .
obs. time interval:  (time[_n-1], time]
 enter on or after:  time time0
 exit on or before:  time futime
------------------------------------------------------------------------------
      178  total obs.
        0  exclusions
------------------------------------------------------------------------------
      178  obs. remaining, representing
       85  subjects
      112  failures in multiple failure-per-subject data
     2480  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =        59
```

and we estimate the Andersen-and-Gill Cox model as

```
. stcox group size number, nohr efron robust nolog
Cox regression -- Efron method for ties
No. of subjects =           85                     Number of obs   =       178
No. of failures =          112
Time at risk    =         2480
                                                   Wald chi2(3)    =     11.41
Log likelihood  =   -449.98064                     Prob > chi2     =    0.0097
                              (standard errors adjusted for clustering on id)
------------------------------------------------------------------------------
      _t |               Robust
      _d |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
   group |   -.464687   .2671369     -1.740   0.082      -.9882656    .0588917
    size |  -.0436603   .0780767     -0.559   0.576      -.1966879    .1093673
  number |   .1749604   .0634147      2.759   0.006       .0506699    .2992509
------------------------------------------------------------------------------
```

This time it was necessary to specify the `cluster()` option. Because `stset`'s `id()` option was used, Stata knows to cluster on the `id()` variable when producing robust standard errors.

3.2.2 *The marginal risk set model (Wei, Lin, and Weissfeld)*

The setup for the marginal risk model is identical to the competing risk model described in section 3.1.2. In essence the model ignores the ordering of events and treats each failure occurrence as belonging in an independent stratum.

The resulting data for the first six of the nine subjects listed above are

```
. list id  group time status number size  rec if id<7, noobs nod
      id      group       time     status     number       size        rec
       1    placebo          1          0          1          3          1
       1    placebo          1          0          1          3          2
       1    placebo          1          0          1          3          3
       1    placebo          1          0          1          3          4
       2    placebo          4          0          2          1          1
       2    placebo          4          0          2          1          2
       2    placebo          4          0          2          1          3
       2    placebo          4          0          2          1          4
       3    placebo          7          0          1          1          1
       3    placebo          7          0          1          1          2
       3    placebo          7          0          1          1          3
       3    placebo          7          0          1          1          4
       4    placebo         10          0          5          1          1
       4    placebo         10          0          5          1          2
       4    placebo         10          0          5          1          3
       4    placebo         10          0          5          1          4
       5    placebo          6          1          4          1          1
       5    placebo         10          0          4          1          2
       5    placebo         10          0          4          1          3
       5    placebo         10          0          4          1          4
       6    placebo         14          0          1          1          1
       6    placebo         14          0          1          1          2
       6    placebo         14          0          1          1          3
       6    placebo         14          0          1          1          4
```

The data is then `stset` without specifying the `id()` option

```
. stset time, failure(status)

     failure event:  status ~= 0 & status ~= .
obs. time interval:  (0, time]
 exit on or before:  failure
------------------------------------------------------------------------------
      340  total obs.
        0  exclusions
------------------------------------------------------------------------------
      340  obs. remaining, representing
      112  failures in single record/single failure data
     8522  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =        59
```

and the Cox model is estimated by clustering on id and stratifying on the failure occurrence variable (`rec`).

```
. stcox group size number,nohr efron strata(rec) cluster(id) nolog

Stratified Cox regr. -- Efron method for ties

No. of subjects = 340                               Number of obs   =       340
No. of failures = 112
Time at risk    = 8522
                                                    Wald chi2(3)    =     15.35
Log likelihood = -426.14683                         Prob > chi2     =    0.0015

                              (standard errors adjusted for clustering on id)
------------------------------------------------------------------------------
      _t |               Robust
      _d |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
   group |  -.5847935   .3097738     -1.888   0.059      -1.191939    .0223521
    size |   -.051617    .095148     -0.542   0.587      -.2381036    .1348697
  number |   .2102937   .0670372      3.137   0.002       .0789032    .3416842
------------------------------------------------------------------------------
                                                                Stratified by rec
```

3.2.3 *The conditional risk set model (time from entry)*

As previously mentioned, there are two variations of the conditional risk set model. The first variation in which time to each event is measured from entry is illustrated in this section.

The data is set up as for Andersen and Gill's method, however, a variable indicating the failure order is included.

The resulting observations for the first nine subjects are

```
. list id if id<10,noobs nod
    id     group      time0       time     status  number    size        str
     1   placebo          0          1          0       1       3          1
     2   placebo          0          4          0       2       1          1
     3   placebo          0          7          0       1       1          1
     4   placebo          0         10          0       5       1          1
     5   placebo          0          6          1       4       1          1
     5   placebo          6         10          0       4       1          2
     6   placebo          0         14          0       1       1          1
     7   placebo          0         18          0       1       1          1
     8   placebo          0          5          1       1       3          1
     8   placebo          5         18          0       1       3          2
     9   placebo          0         12          1       1       1          1
     9   placebo         12         16          1       1       1          2
     9   placebo         16         18          0       1       1          3
```

The resulting dataset is identical to that used to fit Andersen and Gill's model except that the `str` variable identifies the failure risk group for each time span. For the first 4 individuals, who have not had a tumor recur, the `str` value is one, meaning that during their total observed time they are at risk of first failure. The last individual listed, `id`=9, was at risk of a first recurrence for 12 months (`str`=1), at risk of a second recurrence from 12 through 16 months (`str`=2), and at risk of a third recurrence from 16 months to the end of follow-up (`str`=3).

The `stset` command is identical to that used for the Andersen and Gill model.

```
. stset time,fail(status) exit(futime) id(id) enter(time0)
                id:  id
     failure event:  status ~= 0 & status ~= .
obs. time interval:  (time[_n-1], time]
 enter on or after:  time time0
 exit on or before:  time futime
------------------------------------------------------------------------------
      178  total obs.
        0  exclusions
------------------------------------------------------------------------------
      178  obs. remaining, representing
       85  subjects
      112  failures in multiple failure-per-subject data
     2480  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =        59
```

The corresponding conditional risk model is

```
. stcox group size number,nohr efron robust nolog strata(strata)
Stratified Cox regr. -- Efron method for ties
No. of subjects =           85                     Number of obs   =       178
No. of failures =          112
Time at risk    =         2480
                                                   Wald chi2(3)    =      7.17
Log likelihood  =   -315.99082                     Prob > chi2     =    0.0665
                              (standard errors adjusted for clustering on id)
------------------------------------------------------------------------------
      _t |               Robust
      _d |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
   group |  -.3334887   .2060021     -1.619   0.105      -.7372455     .070268
    size |  -.0084947    .062001     -0.137   0.891      -.1300144    .1130251
  number |   .1196172   .0516917      2.314   0.021       .0183033    .2209311
------------------------------------------------------------------------------
                                                      Stratified by strata
```

3.2.4 *The conditional risk set model (time from the previous event)*

The second variations of the conditional risk set model, measures time to each event from the time of the previous event.

The data is set up as in 3.2.3, except that time is not measured continuously from study entry, but the clock is set to zero after each failure.

```
. list id if id<10,noobs nod
      id      group      time0       time     status     number       size        str
       1    placebo          0          1          0          1          3          1
       2    placebo          0          4          0          2          1          1
       3    placebo          0          7          0          1          1          1
       4    placebo          0         10          0          5          1          1
       5    placebo          0          4          0          4          1          2
       5    placebo          0          6          1          4          1          1
       6    placebo          0         14          0          1          1          1
       7    placebo          0         18          0          1          1          1
       8    placebo          0          5          1          1          3          1
       8    placebo          0         13          0          1          3          2
       9    placebo          0          2          0          1          1          3
       9    placebo          0          4          1          1          1          2
       9    placebo          0         12          1          1          1          1
```

Note that the initial times for all time spans are set to zero and that the time variable now reflects the length of the time span. After creating the new time variable, the data needs to be `stset` again.

```
. stset time,fail(status) exit(futime) enter(time0)
     failure event:  status ~= 0 & status ~= .
obs. time interval:  (0, time]
 enter on or after:  time time0
 exit on or before:  time futime
```

```
------------------------------------------------------------------------------
      178  total obs.
        0  exclusions
------------------------------------------------------------------------------
      178  obs. remaining, representing
      112  failures in single record/single failure data
     2480  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =        59
```

The corresponding conditional risk model is

```
. stcox group size number,nohr efron robust nolog strata(str) cluster(id)
Stratified Cox regr. -- Efron method for ties
No. of subjects =          178                     Number of obs   =       178
No. of failures =          112
Time at risk    =         2480
                                                   Wald chi2(3)    =     11.70
Log likelihood  =   -358.96849                     Prob > chi2     =    0.0085
                                (standard errors adjusted for clustering on id)
------------------------------------------------------------------------------
      _t |               Robust
      _d |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
   group |  -.2790045   .2169035     -1.286   0.198      -.7041277    .1461186
    size |   .0074151   .0647143      0.115   0.909      -.1194226    .1342528
  number |   .1580459   .0512421      3.084   0.002       .0576133    .2584785
------------------------------------------------------------------------------
                                                        Stratified by strata
```

4. Conclusion

This paper details how Stata can be used to fit variance-corrected models for the analysis of multiple failure-time data. The examples used to illustrate the various approaches, although real, were simple. More complicated datasets, however, containing time-dependent covariates, varying time scales, delayed entry and other complications, can be set up and analyzed following the guidelines illustrated in this paper.

The most important aspect in the implementation of the methods described, is the accurate construction of the dataset for analysis. Care must be taken to correctly code entry and exit times, strata variables and failure/censoring indicators. It is strongly recommended that, after creating the final dataset and before analyzing and reporting results, the data be examined thoroughly. Lists of all representative, and especially complex cases, should be carefully verified. This step, although time consuming and tedious, is indispensable, especially when working with complicated survival data structures.

A second important aspect of the analysis, is the proper use of the `stset` command. Become familiar and have a clear understanding of the `id()`, `origin()`, `enter()` and `time0()` options. Review the output from `stset` and confirm that the final data contains the expected number of observations and failures. Check any records dropped and verify the data, especially the `stset` created variables, by listing and examining observations.

Lastly fit the model using the correct `stcox` options to produce robust standard errors and, if needed, the strata specific baseline hazard.

References

Andersen, P. K. and R. D. Gill. 1982. Cox's regression model for counting processes: A large sample study. *Annals of Statistics* 10: 1100–1120.

Lee, E. W., L. J. Wei, and D. Amato. 1992. Cox-type regression analysis for large number of small groups of correlated failure time observations. In *Survival Analysis, State of the Art*, 237–247. Netherlands: Kluwer Academic Publishers.

Lin, D. Y. 1994. Cox regression analysis of multivariate failure time data: The marginal approach. *Statistics in Medicine* 13: 2233–2247.

Lin, D. Y. and L. J. Wei. 1989. The robust inference for the Cox proportional hazards model. *Journal of the American Statistical Association* 84: 1074–1078.

Lindor, K. D., E. R. Dickson, W. P. Baldus, et al. 1994. Ursodeoxycholic acid in the treatment of primary biliary cirrhosis. *Gastroenterology* 106: 1284–1290.

Prentice, R. L., B. J. Williams, and A. V. Peterson. 1981. On the regression analysis of multivariate failure time data. *Biometrika* 68: 373–379.

Therneau, T. M. 1997. Extending the Cox model. *Proceedings of the First Seattle Symposium in Biostatistics*. New York: Springer-Verlag.

Wei, L. J., D. Y. Lin, and L. Weissfeld. 1989. Regression analysis of multivariate incomplete failure time data by modeling marginal distributions. *Journal of the American Statistical Association* 84: 1065–1073.

zz9	Cumulative Index for STB-43–STB-48

[an] Announcements

[dm] Data Management

[gr] Graphics

[ip] Instruction on Programming

[sbe] Biostatistics & Epidemiology

[sg] General Statistics

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