STATA
TECHNICAL
BULLETIN

July 1998
STB-44

A publication to promote communication among Stata users

Editor

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

Associate Editors

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

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

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

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

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

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

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

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

Contents of this issue

dm59 Collapsing datasets to frequencies

Nicholas J. Cox, University of Durham, UK FAX (011) 44-91-374-2456, n.j.cox@durham.ac.uk

Syntax

`collfreq` *varlist* [*weight*] [`if` *exp*] [`in` *range*] [, `freq(`*string*`)` `zero` `nomiss`]

`fweight`s are allowed.

Options

`freq(`*string*`)` specifies a name for the frequency variable. If not specified, `freq` and `_freq` are tried in turn. The name must be new.

`zero` specifies that combinations with frequency zero are wanted.

`nomiss` specifies that observations with missing values on any of the variables in *varlist* will be dropped. If not specified, all observations possible are used.

Remarks

Given a list of variables, `collfreq` produces a new dataset consisting of all combinations of that list that exist in the data together with a new variable that contains the frequency of each combination.

Sometimes it is desired to collapse a dataset into frequency form. Several observations identical on one or more variables will be replaced by one such observation together with the frequency of the corresponding set of values. For example, in certain generalized linear models the frequency of some combination of values is the response variable, so we need to produce that response variable. The set of covariate values associated with each frequency is sometimes called a covariate class. Such collapsing is reversible for the variables concerned, as the original data could be reconstituted by using `expand` (see [R] **expand**) with the variable containing the frequencies of each covariate class.

The existing general-purpose command `collapse` (see [R] **collapse**) can be used to produce these frequencies, but only with some extra work beforehand (and frequently afterwards). The motivation for `collfreq` is that it produces the desired results more directly and more quickly.

Examples

First let us illustrate the use of `collapse` for this problem. Suppose we wish to collapse the auto data to a set of frequencies of the two variables `rep78`, which takes values 1, 2, 3, 4 and 5, and `foreign`, which takes values labeled ‘Domestic’ and ‘Foreign’. We first generate a variable that is identically 1, and then count (or alternatively sum) that variable over distinct combinations of `rep78` and `foreign`.

```
. gen freq = 1

. collapse (count) freq , by(rep78 foreign)

. list
        rep78   foreign          freq
  1.        1  Domestic             2
  2.        2  Domestic             8
  3.        3  Domestic            27
  4.        3   Foreign             3
  5.        4  Domestic             9
  6.        4   Foreign             9
  7.        5  Domestic             2
  8.        5   Foreign             9
  9.        .  Domestic             4
 10.        .   Foreign             1
```

The main advantage of `collfreq` over `collapse` is that this can be done more directly in one line:

```
. collfreq rep78 foreign
```

By default, `collfreq` uses the variable name `freq` for the new variable containing the frequencies. If `freq` is in use, then `_freq` is tried. If that is also in use, then the user is reminded to specify a new variable name via the `freq( )` option.

There are three further advantages of `collfreq`. First, it is faster than `collapse` for what it does. Second, the `zero` option makes it easier to have explicit zeros for combinations of variables that do not arise in the data. This could be achieved by using `fillin` (see [R] **fillin**):

```
. fillin rep78 foreign
. drop _fillin
. replace freq = 0 if freq == .
(2 real changes made)
. list
        rep78   foreign          freq
  1.        1  Domestic             2
  2.        1   Foreign             0
  3.        2  Domestic             8
  4.        2   Foreign             0
  5.        3  Domestic            27
  6.        3   Foreign             3
  7.        4  Domestic             9
  8.        4   Foreign             9
  9.        5  Domestic             2
 10.        5   Foreign             9
 11.        .  Domestic             4
 12.        .   Foreign             1
```

Once again, however, `collfreq` with the `zero` option is more direct:

```
. collfreq rep78 foreign, zero
```

The final advantage of `collfreq` is that it is easier to ignore observations with missing values. This can be done by using the `nomiss` option. In contrast, with the `collapse` approach, some preparatory action is needed. In the current example, it would be easy to drop the observations with missing values:

```
. drop if rep78 == . | foreign == .
```

With a larger set of variables, a more efficient approach would be to flag such observations using `mark` and `markout` before dropping them. But `mark` and `markout` are likely to be unfamiliar to Stata users who are not Stata programmers, and `collfreq` with the `nomiss` option is a simpler alternative.

`collfreq` destroys the data in memory, as does `collapse`.

sbe19.1 Tests for publication bias in meta-analysis

Thomas J. Steichen, RJRT, FAX 910-741-1430, steicht@rjrt.com
Matthias Egger, University of Bristol, FAX (011) 44-117-928-7325, m.egger@bristol.ac.uk
Jonathan Sterne, UMDS, London, FAX (011) 44-171-955-4877, j.sterne@umds.ac.uk

Modification of the metabias program

This insert documents four changes to the `metabias` program (Steichen 1998). First, the weighted form of the Egger et al. (1997) regression asymmetry test for publication bias has been replaced by the unweighted form. Second, an error has been corrected in the calculation of the asymmetry test p values for individual strata in a stratified analysis. Third, error trapping has been modified to capture or report problem situations more completely and accurately. Fourth, the labeling of the Begg funnel graph has been changed to properly title the axes when the `ci` option is specified. None of these changes affects the program syntax or operation.

The first change was made because, while there is little theoretical justification for the weighted analysis, justification for the unweighted analysis is straightforward. As before, let $(t_i, v_i),\ i = 1, \ldots, k$, be the estimated effect sizes and sample variances from k studies in a meta-analysis. Egger et al. defined the standardized effect size as $t_i^* = t_i/{v_i}^{1/2}$, and the precision as $s^{-1} = 1/{v_i}^{1/2}$. For the unweighted form of the asymmetry test, they fit t^* to s^{-1} using standard linear regression and the equation $t^* = \alpha + \beta s^{-1}$. A significant deviation from zero of the estimated intercept, $\widehat{\alpha}$, is then interpreted as providing evidence of asymmetry in the funnel plot and of publication bias in the sampled data.

Jonathan Sterne (private communication to Matthias Egger) noted that this "unweighted" asymmetry test is merely a reformulation of a standard weighted regression of the original effect sizes, t_i, against their standard errors, ${v_i}^{1/2}$, where the weights are the usual $1/v_i$. It follows then that the "weighted" asymmetry test is merely a weighted regression of the original effect sizes against their standard errors, but with weights $1/{v_i}^2$. This form has no obvious theoretical justification.

We note further that the "unweighted" asymmetry test weights the data in a manner consistent with the weighting of the effect sizes in a typical meta-analysis (i.e., both use the inverse variances). Thus, bias is detected using the same weighting metric as in the meta-analysis.

For these reasons, this insert restricts `metabias` to the unweighted form of the Egger et al. regression asymmetry test for publication bias.

The second change to `metabias` is straightforward. A square root was inadvertently left out of the formula for the p value of the asymmetry test that is calculated for an individual stratum when option `by()` is specified. This formula has been corrected. Users of this program should repeat any stratified analyses they performed with the original program. Please note that unstratified analyses were not affected by this error.

The third change to `metabias` extends the error-trapping capability and reports previously trapped errors more accurately and completely. A noteworthy aspect of this change is the addition of an error trap for the `ci` option. This trap addresses the situation where epidemiological effect estimates and associated error measures are provided to `metabias` as risk (or odds) ratios and corresponding confidence intervals. Unfortunately, if the user failed to specify option `ci` in the previous release, `metabias` assumed that the input was in the default (*theta*, *se_theta*) format and calculated incorrect results. The current release checks for this situation by counting the number of variables on the command line. If more than two variables are specified, `metabias` checks for the presence of option `ci`. If `ci` is not present, `metabias` assumes it was accidentally omitted, displays an appropriate warning message, and proceeds to carry out the analysis as if `ci` had been specified.

Warning: The user should be aware that it remains possible to provide *theta* and its variance, *var_theta*, on the command line without specifying option `var`. This error, unfortunately, *cannot* be trapped and will result in an incorrect analysis. Though only a limited safeguard, the program now explicitly indicates the data input option specified by the user, or alternatively, warns that the default data input form was assumed.

The fourth change to `metabias` has effect only when options `graph(begg)` and `ci` are specified together. `graph(begg)` requests a funnel graph. Option `ci` indicates that the user provided the effect estimates in their exponentiated form, *exp*(*theta*)—usually a risk or odds ratio, and provided the variability measures as confidence intervals, (*ll*, *ul*). Since the funnel graph always plots *theta* against its standard error, `metabias` correctly generated *theta* by taking the log of the effect estimate and correctly calculated *se_theta* from the confidence interval. The error was that the axes of the graph were titled using the variable name (or variable label, if available) and did not acknowledge the log transform. This was both confusing and wrong and is corrected in this release. Now when both `graph(begg)` and `ci` are specified, if the variable name for the effect estimate is `RR`, the y-axis is titled "log[RR]" and the x-axis is titled "s.e. of: log[RR]". If a variable label is provided, it replaces the variable name in these axis titles.

References

Egger, M., G. D. Smith, M. Schneider, and C. Minder. 1997. Bias in meta-analysis detected by a simple, graphical test. *British Medical Journal* 315: 629–634.

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

sbe24 | metan—an alternative meta-analysis command

Michael J. Bradburn, Institute of Health Sciences, Oxford, UK, m.bradburn@icrf.icnet.uk
Jonathan J. Deeks, Institute of Health Sciences, Oxford, UK, j.deeks@icrf.icnet.uk
Douglas G. Altman, Institute of Health Sciences, Oxford, UK, d.altman@icrf.icnet.uk

Background

When several studies are of a similar design, it often makes sense to try to combine the information from them all to gain precision and to investigate consistencies and discrepancies between their results. In recent years there has been a considerable growth of this type of analysis in several fields, and in medical research in particular. In medicine such studies usually relate to controlled trials of therapy, but the same principles apply in any scientific area; for example in epidemiology, psychology, and educational research. The essence of meta-analysis is to obtain a single estimate of the effect of interest (effect size) from some statistic observed in each of several similar studies. All methods of meta-analysis estimate the overall effect by computing a weighted average of the studies' individual estimates of effect.

`metan` provides methods for the meta-analysis of studies with two groups. With binary data, the effect measure can be the difference between proportions (sometimes called the risk difference or absolute risk reduction), the ratio of two proportions (risk ratio or relative risk), or the odds ratio. With continuous data, both observed differences in means or standardized differences in means (effect sizes) can be used. For both binary and continuous data, either fixed effects or random effects models can be fitted (Fleiss 1993). There are also other approaches, including empirical and fully Bayesian methods. Meta-analysis can be extended to other types of data and study designs, but these are not considered here.

As well as the primary pooling analysis, there are secondary analyses that are often performed. One common additional analysis is to test whether there is excess heterogeneity in effects across the studies. There are also several graphs that can be used to supplement the main analysis.

Recently Sharp and Sterne (1997) presented a program to carry out some of the above analyses, and further programs have been submitted to perform various diagnostics and further analyses. The differences between `metan` and these other programs are discussed below.

Data structure

Consider a meta-analysis of k studies. When the studies have a binary outcome, the results of each study can be presented in a 2×2 table (Table 1) giving the numbers of subjects who do or do not experience the event in each of the two groups (here called intervention and control).

Table 1. Binary data

Study i; $1 \leq i \leq k$	Event	No event
Intervention	a_i	b_i
Control	c_i	d_i

If the outcome is a continuous measure, the number of subjects in each of the two groups, their mean response, and the standard deviation of their responses are required to perform meta-analysis (Table 2).

Table 2. Continuous data

Study i; $(1 \leq i \leq k)$	Group size	Mean response	Standard deviation
Intervention	n_{1i}	m_{1i}	sd_{1i}
Control	n_{2i}	m_{2i}	sd_{2i}

Analysis of binary data using fixed effect models

There are two alternative fixed effect analyses. The inverse variance method (sometimes referred to as Woolf's method) computes an average effect by weighting each study's log odds ratio, log relative risk, or risk difference according to the inverse of their sampling variance, such that studies with higher precision (lower variance) are given higher weights. This method uses large sample asymptotic sampling variances, so it may perform poorly for studies with very low or very high event rates or small sample sizes. In other situations, the inverse variance method gives a minimum variance unbiased estimate.

The Mantel–Haenszel method uses an alternative weighting scheme originally derived for analyzing stratified case–control studies. The method was first described for the odds ratio by Mantel and Haenszel (1959) and extended to the relative risk and risk difference by Greenland and Robins (1985). The estimate of the variance of the overall odds ratio was described by Robins, Greenland, and Breslow (1986). These methods are preferable to the inverse variance method as they have been shown to be robust when data are sparse, and give similar estimates to the inverse variance method in other situations. They are the default in the `metan` command. Alternative formulations of the Mantel–Haenszel methods more suited to analyzing stratified case–control studies are available in the `epitab` commands.

Peto proposed an assumption free method for estimating an overall odds ratio from the results of several large clinical trials (Yusuf, Peto, et al. 1985). The method sums across all studies the difference between the observed (O[a_i]) and expected (E[a_i]) numbers of events in the intervention group (the expected number of events being estimated under the null hypothesis of no treatment effect). The expected value of the sum of O $-$ E under the null hypothesis is zero. The overall log odds ratio is estimated from the ratio of the sum of the O $-$ E and the sum of the hypergeometric variances from individual trials. This method gives valid estimates when combining large balanced trials with small treatment effects, but has been shown to give biased estimates in other situations (Greenland and Salvan 1990).

If a study's 2×2 table contains one or more zero cells, then computational difficulties may be encountered in both the inverse variance and the Mantel–Haenszel methods. These can be overcome by adding a standard correction of 0.5 to all cells in the 2×2 table, and this is the approach adopted here. However, when there are no events in one whole column of the 2×2 table (i.e., all subjects have the same outcome regardless of group), the odds ratio and the relative risk cannot be estimated, and the study is given zero weight in the meta-analysis. Such trials are included in the risk difference methods as they are informative that the difference in risk is small.

Analysis of continuous data using fixed effect models

The weighted mean difference meta-analysis combines the differences between the means of intervention and control groups ($m_{1i} - m_{2i}$) to estimate the overall mean difference (Sinclair and Bracken 1992). A prerequisite of this method is that the response is measured in the same units using comparable devices in all studies. Studies are weighted using the inverse of the variance of the differences in means. Normality within trial arms is assumed, and between trial variations in standard deviations are attributed to differences in precision, and are assumed equal in both study arms.

An alternative approach is to pool standardized differences in means, calculated as the ratio of the observed difference in means to an estimate of the standard deviation of the response. This approach is especially appropriate when studies measure the same concept (e.g., pain or depression) but use a variety of continuous scales. By standardization, the study results are transformed to a common scale (standard deviation units) that facilitates pooling. There are various methods for computing the standardized study results: Glass's method (Glass, et al. 1981) divides the differences in means by the control group standard deviation, whereas Cohen's and Hedges' methods use the same basic approach, but divide by an estimate of the standard deviation obtained from pooling the standard deviations from both experimental and control groups (Rosenthal 1994). Hedges' method incorporates a small sample bias correction factor (Hedges and Olkin 1985). An inverse variance weighting method is used in all the formulations. Normality within trial arms is assumed, and all differences in standard deviations between trials are attributed to variations in the scale of measurement.

Test for heterogeneity

For all the above methods, the consistency or homogeneity of the study results can be assessed by considering an appropriately weighted sum of the differences between the k individual study results and the overall estimate. The test statistic has a χ^2 distribution with $k-1$ degrees of freedom (DerSimonian and Laird 1986).

Analysis of binary or continuous data using random effect models

An approach developed by DerSimonian and Laird (1986) can be used to perform random effect meta-analysis for all the effect measures discussed above (except the Peto method). Such models assume that the treatment effects observed in the trials are a random sample from a distribution of treatment effects with a variance τ^2. This is in contrast to the fixed effect models which assume that the observed treatment effects are all estimates of a single treatment effect. The DerSimonian and Laird methods incorporate an estimate of the between-study variation τ^2 into both the study weights (which are the inverse of the sum of the individual sampling variance and the between studies variance τ^2) and the standard error of the estimate of the common effect. Where there are computational problems for binary data due to zero cells the same approach is used as for fixed effect models.

Where there is excess variability (heterogeneity) between study results, random effect models typically produce more conservative estimates of the significance of the treatment effect (i.e., a wider confidence interval) than fixed effect models. As they give proportionately higher weights to smaller studies and lower weights to larger studies than fixed effect analyses, there may also be differences between fixed and random models in the estimate of the treatment effect.

Tests of overall effect

For all analyses, the significance of the overall effect is calculated by computing a z score as the ratio of the overall effect to its standard error and comparing it with the standard normal distribution. Alternatively, for the Mantel–Haenszel odds ratio and Peto odds ratio method, χ^2 tests of overall effect are available (Breslow and Day 1980).

Graphical analyses

Three plots are available in these programs. The most common graphical display to accompany a meta-analysis shows horizontal lines for each study, depicting estimates and confidence intervals, commonly called a forest plot. The size of the plotting symbol for the point estimate in each study is proportional to the weight that each trial contributes in the meta-analysis. The overall estimate and confidence interval are marked by a diamond. For binary data, a L'Abbé plot (L'Abbé et al. 1987) plots the event rates in control and experimental groups by study. For all data types a funnel plot shows the relation between the effect size and precision of the estimate. It can be used to examine whether there is asymmetry suggesting possible publication bias (Egger et al. 1997), which usually occurs where studies with negative results are less likely to be published than studies with positive results.

Each trial i should be allocated one row in the dataset. There are three commands for invoking the routines; `metan`, `funnel`, and `labbe`, which are detailed below.

Syntax for metan

`metan` *varlist* [`if` *exp*] [`in` *range*] [`,` *options*]

This main meta-analysis routine requires either four or six variables to be declared. When four variables are specified, analysis of binary data is performed. When six, the data are assumed continuous. Following the syntax of Tables 1 and 2, the *varlist* should be either

a b c d

or

n1 m1 sd1 n2 m2 sd2

Scaling and pooling options for metan

Options for binary data

`rr` pool risk ratios (the default).

`or` pool odds ratios.

`rd` pool risk differences.

`fixed` specifies a fixed effect model using the method of Mantel and Haenszel (the default).

`fixedi` specifies a fixed effect model using the inverse variance method.

`peto` specifies that Peto's assumption free method is used to pool odds ratios.

`random` specifies a random effect model using the method of DerSimonian and Laird, with the estimate of heterogeneity being taken from the Mantel–Haenszel model.

`randomi` specifies a random effect model using the method of DerSimonian and Laird, with the estimate of heterogeneity being taken from the inverse variance fixed effect model.

`cornfield` computes confidence intervals for odds ratios by the Cornfield method, rather than the (default) Woolf method.

`chi2` displays the chi-squared statistic (instead of z) for the test of significance of the pooled effect size. This is available only for odds ratios pooled using Peto or Mantel–Haenszel methods.

Options for continuous data

`cohen` pools standardized mean differences by the method of Cohen (the default).

`hedges` pools standardized mean differences by the method of Hedges.

`glass` pools standardized mean differences by the method of Glass.

`nostandard` pools unstandardized mean differences.

`fixed` specifies a fixed effect model using the inverse variance method (the default).

`random` specifies a random effect model using the DerSimonian and Laird method.

General output options for metan

`ilevel()` specifies the significance level (e.g., 90, 95, 99) for the individual trial confidence intervals.

`olevel()` specifies the significance level (e.g., 90, 95, 99) for the overall (pooled) confidence intervals.

`ilevel` and `olevel` need not be the same, and by default are equal to the significance level specified using `set level`.

`sortby()` sorts by given variable(s).

`label([`*namevar*=*variable containing name string*`] [,`*yearvar*=*variable containing year string*`])` labels the data by its name, year, or both. However, neither variable is required. For the table display, the overall length of the label is restricted to 16 characters.

`nokeep` denotes that Stata is not to retain the study parameters in permanent variables (see *Saved results from metan* below).

`notable` prevents the display of the table of results.

`nograph` prevents the display of the graph.

Graphical display options for forest plot in metan

`xlabel()` defines x-axis labels.

`force()` forces the x-axis scale to be in the range specified in `xlabel()`.

`boxsha()` controls box shading intensity, between 0 and 4. The default is 4, which produces a filled box.

`boxsca()` controls box size, which by default is 1.

`texts()` specifies font size for text display on graph. The default size is 1.

`saving(`*filename*`)` saves the forest plot to the specified file.

`nowt` prevents the display of study weight on the graph.

`nostats` prevents the display of study statistics on the graph.

`nooverall` prevents the display of overall effect size on the graph (automatically enforces the `nowt` option).

`t1()`, `t2()`, `b1()` add titles to the graph in the usual manner.

Note that for graphs on the log scale (that is, ORs or RRs), values outside the range $[\,10^{-8}, 10^{8}\,]$ are not displayed. A confidence interval which extends beyond this will have an arrow added at the end of the range; should the effect size and confidence interval be completely off this scale, they will be represented as an arrow.

Saved results from metan

The following results are stored in global macros:

`$S_1`	pooled effect size (ES)	`$S_7`	χ^2 test for heterogeneity
`$S_2`	standard error of ES	`$S_8`	degrees of freedom (χ^2 heterogeneity)
`$S_3`	lower confidence limit of pooled ES	`$S_9`	p(χ^2 heterogeneity)
`$S_4`	upper confidence limit of pooled ES	`$S_10`	χ^2 value for ES (OR only)
`$S_5`	z value for ES	`$S_11`	p(χ^2 for ES) (OR only)
`$S_6`	p(Z)	`$S_12`	estimate of τ^2, between study variance (D&L only)

Also, the following variables are added to the dataset by default (to override this use the `nokeep` option):

Variable name	Definition
`_ES`	Effect size (ES)
`_seES`	Standard error of ES
`_LCI`	Lower confidence limit for ES
`_UCI`	Upper confidence limit for ES
`_WT`	Study weight
`_SS`	Study sample size

Syntax for funnel

`funnel` [*precision_var effect_size*] [`if` *exp*] [`in` *range*] [, *options*]

If the `funnel` command is invoked following `metan` with no parameters specified it will produce a standard funnel plot of precision (1/SE) against treatment effect. Addition of the `noinvert` option will produce a plot of standard error against treatment effect. The alternative sample size version of the funnel plot can be obtained by using the `sample` option (this automatically selects the `noinvert` option). Alternative plots can be created by specifying *precision_var* and *effect_size*. If the effect size is a relative risk or odds ratio, then the `xlog` graph option should be used to create a symmetrical plot.

Options for funnel

All options for `graph` are valid. Additionally, the following may be specified:

`sample` denotes that the y-axis is the sample size and not a standard error.

`noinvert` prevents the values of the precision variable from being inverted.

`ysqrt` represents the y-axis on a square-root scale.

`overall(`*x*`)` draws a dashed vertical line at the overall effect size given by *x*.

Syntax for labbe

`labbe` *a b c d* [`if` *exp*] [`in` *range*] [`weight=`*weightvar*] [, *options*]

Options for labbe

By default, the size of the plotting symbol is proportional to the sample size of the study. If `weight` is specified, the plotting size will be proportional to *weightvar*. All options for `graph` are valid. Additionally, the following two options may be used:

`nowt` declares that the plotted data points are to be the same size.

`percent` displays the event rates as percentages rather than proportions.

One note of caution: depending on the size of the studies, you may need to rescale the graph (using the `psize()` graph option).

There are differences between `metan` and `meta` (Sharp and Sterne 1998). First, `metan` requires a more straightforward data format than `meta`: `meta` requires calculation of the effect size and its standard error (or confidence interval) for each trial, whilst `metan` calculates effect sizes from 2 × 2 tables for binary data, and from means, standard deviations, and samples sizes for continuous data. All commonly used effect sizes (including standardized effect sizes for continuous data) are available as options in metan. Secondly, where meta provides inverse variance, empirical Bayes and DerSimonian and Laird methods for pooling individual studies, metan additionally provides the commonly used Mantel–Haenszel and Peto methods (but does not provide an empirical Bayes method). There are also differences in the format and options for the forest plot.

Example 1: Interventions in smoking cessation

Silagy and Ketteridge (1997) reported a systematic review of randomized controlled trials investigating the effects of physician advice on smoking cessation. In their review, they considered a meta-analysis of trials which have randomized individuals to receive either a minimal smoking cessation intervention from their family doctor or no intervention. An intervention was considered to be "minimal" if it consisted of advice provided by a physician during a single consultation lasting less than 20 minutes (possibly in combination with an information leaflet) with at most one follow-up visit. The outcome of interest was cessation of smoking. The data are presented below:

```
. describe

Contains data from example1.dta
  obs:            16
 vars:             6                            20 Nov 1997 14:58
 size:           544 (95.8% of memory free)
-------------------------------------------------------------------------------
   1. name        str10  %10s
   2. year        float  %9.0g
   3. a           float  %9.0g
   4. r1          float  %9.0g
   5. c           float  %9.0g
   6. r2          float  %9.0g
-------------------------------------------------------------------------------
Sorted by:

. list

           name       year           a         r1          c         r2
  1.      Slama       1990           1        104          1        106
  2.     Porter       1972           5        101          4         90
  3.     Demers       1990          15        292          5        292
  4.    Stewart       1982          11        504          4        187
  5.       Page       1986           8        114          5         68
  6.      Slama       1995          42       2199          5        929
  7.       Haug       1994          20        154          7        109
  8.    Russell       1979          34       1031          8       1107
  9.     Wilson       1982          21        106         11        105
 10.   McDowell       1985          12         85         11         78
 11.       Janz       1987          28        144         12        106
 12.     Wilson       1990          43        577         17        532
 13.     Vetter       1990          34        237         20        234
 14.    Higashi       1995          53        468         35        489
 15.    Russell       1983          43        761         35        659
 16.   Jamrozik       1984          77        512         58        549
```

We start by producing the data in the format of Table 1, and pooling risk ratios by the Mantel–Haenszel fixed effect method.

```
. gen b=r1-a

. gen d=r2-c

.  metan a b c d, rr label(namevar=name,yearvar=year) xlabel(0.1,0.2,0.5,1,2,5
> ,10) force texts(1.25) t1(Impact of physician advice in) t2(smoking cessation)

           Study |       RR   [95% Conf. Interval]    % Weight
-----------------+-------------------------------------------------------
Slama (1990)     |  1.01923     .064601   16.0808      .400756
Porter (1972)    |  1.11386     .308545   4.02109      1.71164
Demers (1990)    |        3      1.1047   8.14701      2.02305
Stewart (1982)   |  1.02034     .328944   3.16494       2.3609
Page (1986)      |  .954386     .325327    2.7998      2.53436
Slama (1995)     |   3.5487     1.40851   8.94089      2.84442
Haug (1994)      |  2.02226     .886205   4.61468      3.31687
Russell (1979)   |  4.56329     2.12242   9.81126      3.12181
Wilson (1982)    |  1.89108     .960226   3.72432      4.47179
McDowell (1985)  |  1.00107     .468999   2.13676      4.64183
Janz (1987)      |  1.71759     .916596   3.21857      5.59332
Wilson (1990)    |  2.33214     1.34685   4.03825      7.15746
Vetter (1990)    |  1.67848     .995947   2.82876      8.14372
Higashi (1995)   |  1.58223     1.05253   2.37852      13.8506
Russell (1983)   |   1.0639     .689426   1.64178      15.1785
Jamrozik (1984)  |  1.42353     1.03495   1.95799       22.649
```

```
-----------------+-----------------------------------------------------
  M-H pooled RR  |  1.67635     1.44004   1.95145
-----------------+-----------------------------------------------------
  Heterogeneity chi-squared =  21.51 (d.f. = 15) p = 0.121
  Test of RR=1 : z= 6.66 p = 0.000
```

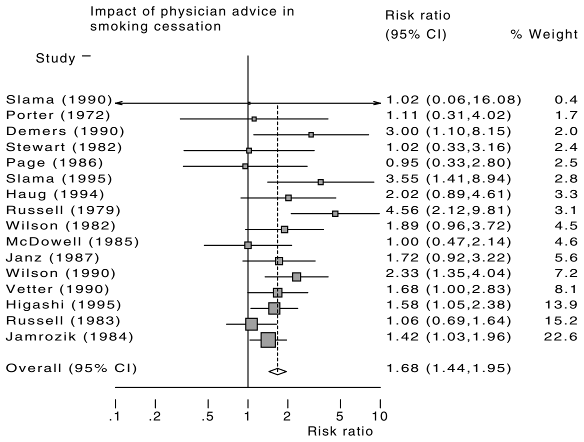

Figure 1. Forest plot for Example 1.

It appears that there is a significant benefit of such minimal intervention. The nonsignificance of the test for heterogeneity suggests that the differences between the studies are explicable by random variation, although this test has low statistical power. The L'Abbé plot provides an alternative way of displaying the data which allows inspection of the variability in experimental and control group event rates.

```
. labbe a b c d , xlabel(0,0.1,0.2,0.3) ylabel(0,0.1,0.2,0.3) psize(50) t1(Impact of physician
>  advice in smoking cessation:) t2(Proportion of patients ceasing to smoke) l1(Physician
>  intervention group patients) b2(Control group patients)
```
(*See Figure 2 below*)

A funnel plot can be used to investigate the possibility that the studies which were included in the review were a biased selection. The alternative command `metabias` (Steichen 1998) additionally gives a formal test for nonrandom inclusion of studies in the review.

```
. funnel , xlog ylabel(0,2,4,6) xlabel(0.5,1,2,5) xli(1) overall(1.68) b2(Risk Ratio)
```
(*See Figure 3 below*)

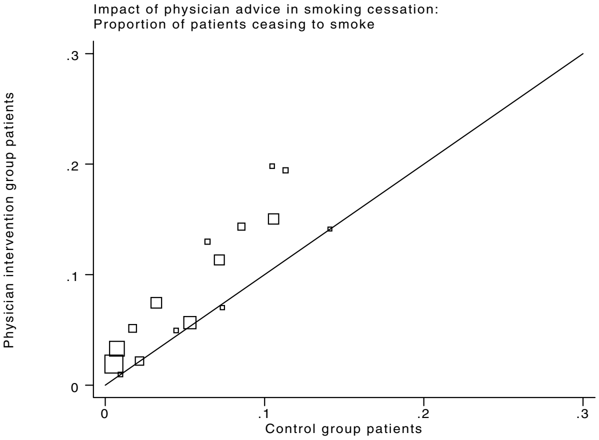

Figure 2. L'Abbé plot for Example 1.

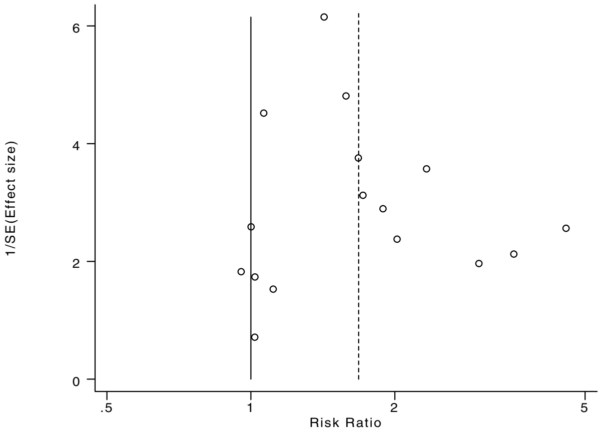

Figure 3. Funnel plot for Example 1.

Interpretation of funnel plots can be difficult, as a certain degree of asymmetry is to be expected by chance.

Example 2

D'Agostino and Weintraub (1995) reported a meta-analysis of the effects of antihistamines in common cold preparations on the severity of sneezing and runny nose. They combined data from nine randomized trials in which participants with new colds were randomly assigned to an active antihistamine treatment or placebo. The effect of the treatment was measured as the change in severity of runny nose following one day's treatment. The trials used a variety of scales for measuring severity. Due to this, standardized mean differences are used in the analysis. We choose to use Cohen's method (the default) to compute the standardized mean difference.

```
. use example2

. list n1 mean1 sd1 n2 mean2 sd2

           n1      mean1        sd1        n2      mean2        sd2
  1.       11       .273       .786        16      -.188       .834
  2.      128       .932       .593       136        .81       .556
  3.       63        .73       .745        64       .578       .773
  4.       22        .35      1.139        22       .339       .744
  5.       16       .422      2.209        15       -.17      1.374
  6.       39       .256      1.666        41       .537      1.614
  7.       21      2.831      1.753        21      1.396      1.285
  8.       13      2.687      1.607         8      1.625      2.089
  9.      194        .49       .895       193       .264       .828

. metan n1 mean1 sd1 n2 mean2 sd2, xlabel(-1.5,-1,-0.5,0,0.5,1,1.5)
> t1(Effect of antihistamines on cold severity)
           Study |      SMD   [95% Conf. Interval]    % Weight
-----------------+-----------------------------------------------------
1                |  .565548    -.217963   1.34906      2.48229
2                |  .212457    -.029593   .454508      26.0094
3                |    .2002    -.148532   .548932      12.5302
4                |  .011435    -.579522   .602391      4.36346
5                |  .236853    -.470186   .943892      3.04828
6                |  -.17139    -.610612   .267833      7.89901
7                |  .933687     .295162   1.57221      3.73754
8                |  .590111    -.310379    1.4906      1.87925
9                |  .262109     .061989   .462229      38.0506
-----------------+-----------------------------------------------------
  I-V pooled SMD |  .234284      .11084   .357728
-----------------+-----------------------------------------------------
  Heterogeneity chi-squared =   9.86 (d.f. = 8) p = 0.275
  Test of SMD=0 : z= 3.72 p = 0.000
```

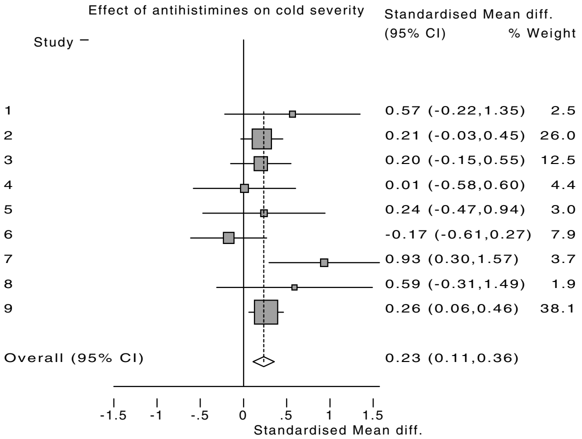

Figure 4. Forest plot for Example 2.

The patients given antihistamines appear to have a greater reduction in severity of cold symptoms in the first 24 hours of treatment. Again the between-study differences are explicable by random variation.

Formulas

Individual study responses: binary outcomes

For study i denote the cell counts as in Table 1, and let $n_{1i} = a_i + b_i$, $n_{2i} = c_i + d_i$ (the number of participants in the treatment and control groups respectively) and $N_i = n_{1i} + n_{2i}$ (the number in the study). For the Peto method the individual odds ratios are given by

$$\widehat{OR}_i = \exp\left\{\left(a_i - \mathrm{E}\left[a_i\right]\right)/v_i\right\}$$

with its logarithm having standard error

$$\mathrm{se}\{\ln(\widehat{OR}_i)\} = \sqrt{1/v_i}$$

where $\mathrm{E}[a_i] = n_{1i}(a_i + c_i)/N_i$ (the expected number of events in the exposure group) and

$v_i = [n_{1i}n_{2i}(a_i + c_i)(b_i + d_i)]/[N_i^2(N_i - 1)]$ (the hypergeometric variance of a_i).

For other methods of combining trials, the odds ratio for each study is given by

$$\widehat{OR}_i = a_i d_i / b_i c_i$$

the standard error of the log odds ratio being

$$\mathrm{se}\{\ln(\widehat{OR}_i)\} = \sqrt{1/a_i + 1/b_i + 1/c_i + 1/d_i}$$

The risk ratio for each study is given by

$$\widehat{RR}_i = (a_i/n_{1i})/(c_i/n_{2i})$$

the standard error of the log risk ratio being

$$\mathrm{se}\{\ln(\widehat{RR}_i)\} = \sqrt{1/a_i + 1/c_i - 1/n_{1i} - 1/n_{2i}}$$

The risk difference for each study is given by

$$\widehat{RD}_i = (a_i/n_{1i}) - (c_i/n_{2i}) \text{ with standard error } \mathrm{se}(\widehat{RD}_i) = \sqrt{a_i b_i/n_{1i}^3 + c_i d_i/n_{2i}^3}$$

where zero cells cause problems with computation of the standard errors, 0.5 is added to all cells (a_i,b_i,c_i,d_i) for that study.

Individual study responses: continuous outcomes

Denote the number of subjects, mean and standard deviation as in Table 1, and let

$$N_i = n_{1i} + n_{2i}$$

and

$$s_i = \sqrt{((n_{1i} - 1)sd_{1i}^2 + (n_{2i} - 1)sd_{2i}^2)/(N_i - 2)}$$

be the pooled standard deviation of the two groups. The weighted mean difference is given by

$$\widehat{WMD}_i = m_{1i} - m_{2i} \text{ with standard error } \mathrm{se}(\widehat{WMD}_i) = \sqrt{sd_{1i}^2/n_{1i} + sd_{2i}^2/n_{2i}}$$

There are three formulations of the standardized mean difference. The default is the measure suggested by Cohen (Cohen's d) which is the ratio of the mean difference to the pooled standard deviation s_i; i.e.,

$$\widehat{d}_i = (m_{1i} - m_{2i})/s_i \text{ with standard error } \mathrm{se}(\widehat{d}_i) = \sqrt{N_i/(n_{1i}n_{2i}) + \widehat{d}_i^2/2(N_i - 2)}$$

Hedges suggested a small-sample adjustment to the mean difference (Hedges adjusted g), to give

$$\widehat{g}_i = ((m_{1i} - m_{2i})/s_i)(1 - 3/(4N_i - 9)) \text{ with standard error } \mathrm{se}(\widehat{g}_i) = \sqrt{N_i/(n_{1i}n_{2i}) + \widehat{g}_i^2/2(N_i - 3.94)}$$

Glass suggested using the control group standard deviation as the best estimate of the scaling factor to give the summary measure (Glass's $\widehat{\Delta}$), where

$$\widehat{\Delta}_i = (m_{1i} - m_{2i})/sd_{2i}, \text{ with standard error } \mathrm{se}(\Delta_i) = \sqrt{N_i/(n_{1i}n_{2i}) + \widehat{\Delta}_i^2/2(n_{2i} - 1)}$$

Mantel–Haenszel methods for combining trials

For each study, the effect size from each trial $\widehat{\theta}_i$ is given weight w_i in the analysis. The overall estimate of the pooled effect, $\widehat{\theta}_{MH}$ is given by

$$\widehat{\theta}_{MH}=(\sum w_i\widehat{\theta}_i)/(\sum w_i)$$

For combining odds ratios, each study's OR is given weight

$$w_i = b_i c_i/N_i,$$

and the logarithm of $\widehat{OR}_{MH}$ has standard error given by

$$\mathrm{se}\{\ln(\widehat{OR}_{MH})\} = \sqrt{(PR)/2R^2 + ((PS + QR)/2(R \times S)) + (QS)/2S^2}$$

where

$$R = \sum a_i d_i/N_i \qquad S = \sum b_i c_i/N_i$$

$$PR = \sum(a_i + d_i)a_i d_i/N_i^2 \qquad PS = \sum(a_i + d_i)b_i c_i/N_i^2$$

$$QR = \sum(b_i + c_i)a_i d_i/N_i^2 \qquad QS = \sum(b_i + c_i)b_i c_i/N_i^2$$

For combining risk ratios, each study's RR is given weight

$$w_i = (n_{1i}c_i)/N_i$$

and the logarithm of $\widehat{RR}_{MH}$ has standard error given by

$$\text{se}\{\ln(\widehat{RR}_{MH})\} = \sqrt{P/(R \times S)}$$

where

$$P = \sum(n_{1i}n_{2i}(a_i + c_i) - a_ic_iN_i)/N_i^2 \qquad R = \sum a_in_{2i}/N_i \qquad S = \sum c_in_{1i}/N_i$$

For risk differences, each study's RD has the weight

$$w_i = n_{1i}n_{2i}/N_i$$

and $\widehat{RD}_{MH}$ has standard error given by

$$\text{se}\{\widehat{RD}_{MH}\} = \sqrt{(P/Q^2)}$$

where

$$P = \sum(a_ib_in_{2i}^3 + c_id_in_{1i}^3)/n_{1i}n_{2i}N_i^2; \qquad Q = \sum n_{1i}n_{2i}/N_i$$

The heterogeneity statistic is given by

$$Q = \sum w_i(\widehat{\theta}_i - \widehat{\theta}_{MH})^2$$

where θ is the log odds ratio, log relative risk or risk difference. Under the null hypothesis that there are no differences in treatment effect between trials, this follows a χ^2 distribution on $k - 1$ degrees of freedom.

Inverse variance methods for combining trials

Here, when considering odds ratios or risk ratios, we define the effect size θ_i to be the natural logarithm of the trial's OR or RR; otherwise, we consider the summary statistic (RD, SMD or WMD) itself. The individual effect sizes are weighted according to the reciprocal of their variance (calculated as the square of the standard errors given in the individual study section above) giving

$$w_i = 1/\text{se}(\widehat{\theta}_i)^2$$

These are combined to give a pooled estimate

$$\widehat{\theta}_{IV} = (\sum w_i\widehat{\theta}_i)/(\sum w_i)$$

with

$$\text{se}\{\widehat{\theta}_{IV}\} = 1/\sqrt{\sum w_i}$$

The heterogeneity statistic is given by a similar formula as for the Mantel–Haenszel method, using the inverse variance form of the weights, w_i

$$Q = \sum w_i(\widehat{\theta}_i - \widehat{\theta}_{IV})^2$$

Peto's assumption free method for combining trials

Here, the overall odds ratio is given by

$$\widehat{OR}_{Peto} = \exp\{\sum w_i \ln(\widehat{OR}_i)/\sum w_i\}$$

where the odds ratio $\widehat{OR}_i$ is calculated using the approximate method described in the individual trial section, and the weights, w_i are equal to the hypergeometric variances, v_i.

The logarithm of the odds ratio has standard error

$$\text{se}\{\ln(\widehat{OR}_{Peto})\} = 1/\sqrt{\sum w_i}$$

The heterogeneity statistic is given by

$$Q = \sum w_i\{(\ln \widehat{OR}_i)^2 - (\ln \widehat{OR}_{Peto})^2\}$$

DerSimonian and Laird random effect models

Under the random effect model, the assumption of a common treatment effect is relaxed, and the effect sizes are assumed to have a distribution

$$\theta_i \sim N(\theta, \tau^2)$$

The estimate of τ^2 is given by

$$\widehat{\tau}^2 = \max\{[Q-(k-1)]/[\sum w_i - (\sum(w_i^2)/\sum w_i)], 0\}$$

The estimate of the combined effect for heterogeneity may be taken as either the Mantel–Haenszel or the inverse variance estimate. Again, for odds ratios and risk ratios, the effect size is taken as the natural logarithm of the OR and RR. Each study's effect size is given weight

$$w_i = 1/(\mathrm{se}(\widehat{\theta}_i)^2 + \widehat{\tau}^2)$$

The pooled effect size is given by

$$\widehat{\theta}_{DL} = (\sum w_i \widehat{\theta}_i)/(\sum w_i)$$

and

$$\mathrm{se}\{\widehat{\theta}_{DL}\} = 1/\sqrt{\sum w_i}$$

Note that in the case where the heterogeneity statistic Q is less than or equal to its degrees of freedom $(k-1)$, the estimate of the between trial variation, $\widehat{\tau}^2$, is zero, and the weights reduce to those given by the inverse variance method.

Confidence intervals

The $100(1-\alpha)\%$ confidence interval for $\widehat{\theta}$ is given by

$$\widehat{\theta} - \mathrm{se}(\widehat{\theta})\Phi(1-\alpha/2), \quad \text{to} \quad \widehat{\theta} + \mathrm{se}(\widehat{\theta})\Phi(1-\alpha/2)$$

where $\widehat{\theta}$ is the log odds ratio, log relative risk, risk difference, mean difference or standardized mean difference, and Φ is the standard normal distribution function. The Cornfield confidence intervals for odds ratios are calculated as explained in the Stata manual for the `epitab` command.

Test statistics

In all cases, the test statistic is given by

$$z = \widehat{\theta}/\mathrm{se}(\widehat{\theta})$$

where the odds ratio or risk ratio is again considered on the log scale.

For odds ratios pooled by method of Mantel and Haenszel or Peto, an alternative test statistic is available, which is the χ^2 test of the observed and expected events rate in the exposure group. The expectation and the variance of a_i are as given earlier in the Peto odds ratio section. The test statistic is

$$\chi^2 = \{\sum(a_i - E[a_i])\}^2/\sum \mathrm{var}(a_i)$$

on one degree of freedom. Note that in the case of odds ratios pooled by method of Peto, the two test statistics are identical; the χ^2 test statistic is simply the square of the z score.

Acknowledgments

The statistical methods programmed in `metan` utilize several of the algorithms used by the MetaView software (part of the Cochrane Library), which was developed by Gordon Dooley of Update Software, Oxford and Jonathan Deeks of the Statistical Methods Working Group of the Cochrane Collaboration. We have also used a subroutine written by Patrick Royston of the Royal Postgraduate Medical School, London.

References

Breslow, N. E. and N. E. Day. 1980. Combination of results from a series of 2x2 tables; control of confounding. In *Statistical Methods in Cancer Research*, vol. 1, Lyon: International Agency for Health Research on Cancer.

D'Agostino, R. B. and M. Weintraub. 1995. Meta-analysis: A method for synthesizing research. *Clinical Pharmacology and Therapeutics* 58: 605–616.

DerSimonian, R. and N. Laird. 1986. Meta-analysis in clinical trials. *Controlled Clinical Trials* 7: 177–188.

Egger, M., G. D. Smith, M. Schneider, and C. Minder. 1997. Bias in meta-analysis detected by a simple, graphical test. *British Medical Journal* 315: 629–635.

Fleiss, J. L. 1993. The statistical basis of meta-analysis. *Statistical Methods in Medical Research* 2: 121–145.

Glass, G. V., B. McGaw, and M. L. Smith. 1981. *Meta-Analysis in Social Research*. Beverly Hills, CA: Sage Publications.

Greenland, S. and J. Robins. 1985. Estimation of a common effect parameter from sparse follow-up data. *Biometrics* 41: 55–68.

Greenland, S. and A. Salvan. 1990. Bias in the one-step method for pooling study results. *Statistics in Medicine* 9: 247–252.

Hedges, L. V. and I. Olkin. 1985. *Statistical Methods for Meta-analysis*. San Diego: Academic Press. Chapter 5.

L'Abbé, K. A., A. S. Detsky, and K. O'Rourke. 1987. Meta-analysis in clinical research. *Annals of Internal Medicine* 107: 224–233.

Mantel, N. and W. Haenszel. 1959. Statistical aspects of the analysis of data from retrospective studies of disease. *Journal of the National Cancer Institute* 22: 719–748.

Robins, J., S. Greenland, and N. E. Breslow. 1986. A general estimator for the variance of the Mantel–Haenszel odds ratio. *American Journal of Epidemiology* 124: 719–723.

Rosenthal, R. 1994. Parametric measures of effect size. In *The Handbook of Research Synthesis*, ed. H. Cooper and L. V. Hedges. New York: Russell Sage Foundation.

Sharp, S. and J. Sterne. 1997. sbe16: Meta-analysis. *Stata Technical Bulletin* 38: 9–14. Reprinted in *The Stata Technical Bulletin Reprints* vol. 7, pp. 100–106.

Silagy, C. and S. Ketteridge. 1997. Physician advice for smoking cessation. In *Tobacco Addiction Module of The Cochrane Database of Systematic Reviews*, ed. T. Lancaster, C. Silagy, and D. Fullerton. [updated 01 September 1997]. Available in The Cochrane Library [database on disk and CDROM]. The Cochrane Collaboration; Issue 4. Oxford: Update Software. Updated quarterly.

Sinclair, J. C. and M. B. Bracken. 1992. *Effective Care of the Newborn Infant*. Oxford: Oxford University Press. Chapter 2.

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

Yusuf, S., R. Peto, J. Lewis, R. Collins, and P. Sleight. 1985. Beta blockade during and after myocardial infarction: an overview of the randomized trials. *Progress in Cardiovascular Diseases* 27: 335–371.

sg85 Moving summaries

Nicholas J. Cox, University of Durham, UK, FAX (011) 44-91-374-2456, n.j.cox@durham.ac.uk

Syntax

```
movsumm varname [if exp] [in range] [weight] , gen(newvar) result(#)
        [ window(#) end mid { binomial | oweight(string) } wrap ]
```

`fweight`s and `aweight`s are allowed.

Description

`movsumm` produces a new variable containing moving summaries of *varname* for overlapping windows of specified length. *varname* will usually (but not necessarily) be a time series with regularly spaced values. Possible summaries are those produced by `summarize` and saved in `_result()`.

It is the user's responsibility to place observations in the appropriate sort order first.

Options

`gen(`*newvar*`)` specifies *newvar* as the name for the new variable. It is in fact a required option.

`result(`*#*`)` specifies which `_result()` from `summarize` is to be used. It is in fact a required option. See the table below. Note the typographical error in the Stata 5.0 manual entry [R] **summarize**: `_result(10)` contains the 50th percentile (median).

#	meaning	#	meaning
1	number of observations	10	50th percentile (median)
2	sum of weight	11	75th percentile
3	mean	12	90th percentile
4	variance	13	95th percentile
5	minimum	14	skewness
6	maximum	15	kurtosis
7	5th percentile	16	1st percentile
8	10th percentile	17	99th percentile
9	25th percentile	18	sum of variable

`window(`*#*`)` specifies the length of the window, which should be an integer at least 2. The default is 3. By default, results for odd-length windows are placed in the middle of the window and results for even-length windows are placed at the end of the window. The defaults can be overridden by `end` or `mid`.

`end` forces results to be placed at the end of the window.

`mid` forces results to be placed in the middle of the window, or in the case of windows of even length just after it: in the 2nd of 2, the 3rd of 4, the 4th of 6, and so on.

binomial specifies that binomial weights are used. For a window of length k, these are

$$\frac{\binom{k-1}{i-1}}{2^{k-1}}, \qquad i = 1, \dots, k$$

oweight(*string*) specifies the user's own weights for values in the window. These should be numbers separated by spaces (not commas). The number of weights should equal that specified by window(). Thus ow(0.3 0.4 0.3) specifies weights of 0.3, 0.4, 0.3.

binomial and oweight() are mutually exclusive. Neither may be combined with other weights specified by [w =]. Each is treated as specifying analytic weights, i.e. they are rescaled to sum to the number of observations.

wrap specifies that beginning and end values should be calculated by wrapping around. Thus with window length 3 and 12 observations the window for observation 1 would be 12, 1, 2 and the window for observation 12 would be 11, 12, 1. The justification for this option would be that the data are periodic, for example time of day or time of year.

Explanation

movsumm will principally be useful for processing variables which are time series, such as temperatures or sales measured at or for different points or periods of time. However, it could also be applied to similar one-dimensional spatial series, if instead of time we have say some measure of distance or depth, or more generally to any response that has been sorted according to the value of some covariate. In what follows, it is assumed that the user has observations already sorted according to time or whatever else provides an order. If not, then use sort (see [R] **sort**) beforehand.

Note that movsumm takes no account of the exact spacing of the observations. In most applications, the data will come regularly spaced (for example, every day, month, year). In other applications movsumm may still provide useful results, but it should be remembered that the window may vary in length (in time, space, whatever), even though it contains a fixed number of observations.

In time series analysis, it is common, especially for basic descriptive or exploratory work, to summarize the series within a window which is moved along the series. The summary indicates the state of affairs within the window, that is, locally or over a short period. movsumm is concerned with windows that overlap. For example, if the window is of length 7, the first window consists of observations 1 through 7, the second 2 through 8, and so on. movsumm could also be used for calculations involving disjoint windows, but it is already easy to do most such calculations in Stata. The most basic method is simply to use generate in conjunction with by. Other possibilities are provided under egen.

Most frequently, the summary calculated in each window is some measure of level, such as a mean or median. In particular, various means, whether weighted or not, are called moving averages or running means. More generally, the calculation of the moving measure of level is often called smoothing or filtering. Smoothing is a large area within statistical science and many Stata commands offer some kind of smoothing, but the two closest in spirit to movsumm are provided in egen and smooth.

egen with the ma option calculates equally weighted moving averages for windows with odd numbers of observations. See [R] **egen**. smooth principally offers various nonlinear smoothers based on medians. See [R] **smooth**. smooth also allows one or more applications of Hanning, a linear smoother or moving average with weights $\frac{1}{4} : \frac{1}{2} : \frac{1}{4}$. Those weights are binomial coefficients 1 : 2 : 1, divided by their sum 4, so that they add to 1. By repeated applications of Hanning, any desired binomial smoother can be produced with smooth, as any other binomial coefficients can be obtained by convolution. This fact was exploited in the bsmplot command published in Cox (1997).

Other moving averages could be implemented in Stata by *ad hoc* one-line commands. Suppose you wanted equally-weighted averages of a variable x for windows of length 4. These are not obtainable through the ma option of egen, because 4 is even, not odd.

```
. gen ave4 = (1/4) * (x[_n-3] + x[_n-2] + x[_n-1] + x)
```

is one direct way to do it. Note that there is a minor issue: where should the result be put? In this case, the average is put at the end of the window to which it refers. The new variable ave4 will contain some missing values, in this case at the beginning of the series, because for the first three observations there are fewer than 4 values to use. In some cases, it may be important to fill in missing values in some way, but frequently people do not bother, because the aim of the procedure is just to get an idea of the general structure of the data, and a lack of results at either end of the series is thus of little concern.

Even in the case of Hanning, Stata would be faster at doing

```
. gen bin3 = (1/4) * (x[_n-1] + 2 * x + x[_n+1])
```

than at doing

```
. smooth H x, gen(bin3)
```

or

```
. movsumm x, gen(bin3) bin w(3) r(3)
```

because the first is much more direct than the second or third.

However, even though Stata is typically quicker doing the direct calculation, the difference in time is often much less than the time spent by the user working out the syntax required. Why then use `movsumm`?

First, as a tool for calculating moving averages, whether weighted or unweighted (that is, equally weighted), `movsumm` is much more general than either `egen` with `ma` or `smooth`, given that windows can be any length desired, the result can be put at the end or in the middle of the window, and weights within the window can be specified.

The two last points need expansion. `movsumm` has an idea of the natural place to put its results. If the length of the window is odd (3, 5, 7, etc.), there is a natural midpoint to the window, and that is where `movsumm` puts results by default. If the length of the window is even (2, 4, 6, etc.), there is no such midpoint. In any case, my guess is that most people using even-length windows are likely to be economists and others using monthly or quarterly data and thus windows of length 4 or 12. The accompanying idea is essentially: at the end of the year, or the quarter, how have we done? Hence the default for even-length windows is that `movsumm` puts its results at the end of each window, which seems the natural place. If you disagree with these choices for defaults, you can override them by using either the `end` option or the `mid` option.

`movsumm` offers two possibilities for weights according to position in the window: binomial weights, with the `binomial` option, or the user's own weights, spelled out in the `oweight( )` option. Note that such weights cannot be combined with other weights specified by `[w = ]`. The latter may be `fweight`s or `aweight`s, but weights supplied by `binomial` or `oweight( )` are treated as `aweight`s.

There is one exception to the statement that `movsumm` is more general than its two nearest relatives. `movsumm` does no surgery at either end of the series to replace missing values. If that is important to you, check out those commands, or devise your own surgery.

Second, although it is less common in practice than the calculation of moving averages, other summary statistics may be required for each window. With `movsumm` you can pick up any result left behind by `summarize` in `_result( )`. The full list is given in the explanation of the `result( )` option.

Suppose we want totals of monthly sales in the previous quarter. The sum is left by `summarize` in `_result(18)`, so we type

```
. movsumm monsales, w(4) r(18) gen(qtrsales)
```

Once again, Stata will be slower at doing this than at doing

```
. gen qtrsales = monsales + monsales[_n-1] + monsales[_n-2] + monsales[_n-3]
```

but by the time you have typed the longer command, the difference is immaterial.

The list of results obtainable implies other possibilities. If you wanted a moving standard deviation for windows of length 7, that is two steps:

```
. movsumm x, gen(varx) w(7) r(4)
. gen sdx = sqrt(varx)
```

`summarize` leaves the variance behind in `_result(4)`. If you wanted a moving interquartile range for windows of length 7, that is three steps:

```
. movsumm x, gen(upqx) w(7) r(11)
. movsumm x, gen(loqx) w(7) r(9)
. gen iqrx = upqx - loqx
```

`summarize` leaves the lower and upper quartiles behind in `_result(9)` and `_result(11)`.

Third, `movsumm` allows the special treatment that is often appropriate for periodic variables. If time is time of year or time of day, then that scale is circular. January follows December just as February follows January. Hence it may seem sensible that windows wrap around. The `wrap` option specifies that beginning and end values should be calculated in this way. Thus with a window of 3 and 12 observations, the window for observation 1 would be 12, 1, 2 and the window for observation 12 would be 11, 12, 1. `movsumm` temporarily creates enough extra observations to ensure that the user has a complete set of results.

Suppose that we want binomial smoothing with window length 7, wrapped around:

```
. movsumm temp, w(7) r(3) gen(bin7temp) bin wrap
```

Examples

In practice, the point of using `movsumm` is to create variables for graphing or other analysis. As a real example, we use data from the accompanying file `gltemp.dta`, which contains average temperatures for each year from 1851 to 1997 for the world and the Northern and Southern hemispheres. Each temperature variable is expressed as deviations from the 1961–90 mean, in °C. (In climatology, 30 year averages starting in the first year of a decade are conventional reference levels.) The data were downloaded from `http://www.meto.govt.uk/sec5/CR_div/Tempertr/index.html`.

We generate binomially weighted means and standard deviations for window lengths of 21 years. The binomial weighting means that values get progressively less weight away from the midpoint of each window.

```
. movsumm global, r(3) bin g(bin21) w(21)
. movsumm global, r(4) bin g(var21) w(21)
. gen sd21 = sqrt(var21)
(20 missing values generated)
. graph global bin21 sd21 year , sy(oii) c(.ss) xlab ylab
```

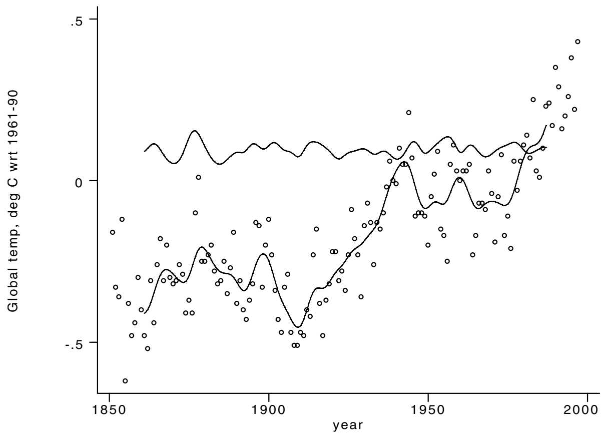

Figure 1

The means and standard deviations are plotted at the center of each window. While changes in the mean are striking, the standard deviation appears to fluctuate around a constant value.

References

Cox, N. J. 1997. gr22: Binomial smoothing plot. *Stata Technical Bulletin* 35: 7–9. Reprinted in *The Stata Technical Bulletin Reprints*, vol. 6, pp. 36–38.

sg86 Continuation-ratio models for ordinal response data

Rory Wolfe, Royal Children's Hospital, Australia, wolfer@cryptic.rch.unimelb.edu.au

Background

In modeling an ordinal outcome (with J categories), it is often reasonable to think of some continuum underlying the categories of the ordinal scale. The observations are then thought of as being crude measures of a response variable that is defined on the underlying continuum but is unmeasurable. Each ordinal observation indicates that the underlying response variable lies between two "cut-points" on the continuum. There are $J-1$ cut-points in total, dividing the continuum into J separate parts, each corresponding to an outcome category.

This approach leads naturally to consideration of models based on cumulative probabilities, $\gamma_j = \mathrm{Pr}(\mathrm{Ordinal\ outcome} \leq \mathrm{Category}\ j)$, two of which can be fitted in Stata 5.0 using the commands `ologit` and `oprobit`. The only difference between the models fitted by these commands is in the link function used to transform the cumulative probabilities to a continuum where they can be modeled by a linear combination of explanatory effects. The former command fits a model that employs the logit link whereas the latter fits a probit-link model. With the new command, `ocratio`, it is possible to fit a third model which uses the complementary-log-log link function.

The primary purpose of introducing `ocratio` is to make available an alternative type of model, based on quantities other than cumulative probabilities, with which to model ordinal data. These models are based on continuation-ratio probabilities

$$\delta_j = \mathrm{Pr}(\mathrm{Ordinal\ outcome} = \mathrm{category}\ j|\mathrm{Outcome\ is\ category}\ j\ \mathrm{or\ greater})$$

and motivation for their use is given by McCullagh and Nelder (1989). To summarize their discussion, consider an ordinal response Y, the number of inseminations required for a milch cow to become pregnant. This outcome is the manifest of a series of binary outcomes, success or failure of artificial insemination, observed on the first, second, third, etc. attempted insemination. Starting with n cows, each with probability π_1 of getting pregnant from the first insemination, suppose we observe a total of y_1 pregnancies at this stage. A second insemination will take place in $n - y_1$ cows and the probability of a pregnancy at this stage is $\pi_2/(1-\pi_1)$ with y_2 observed pregnancies. At the next stage there are $n - y_1 - y_2$ cows with probability $\pi_3/(1-\pi_1-\pi_2)$ of getting pregnant. This sequence continues until all cows are pregnant.

Now consider a binary regression model for each insemination j (assuming all cows are pregnant after the fourth insemination). If the row vector $\mathbf{x}_i$ contains values of explanatory variables for the ith cow and β_j is a parameter vector describing the effect of the explanatory variables on the binary outcome, then we have

$$\begin{aligned} \mathrm{link}(\pi_{i1}) =& \kappa_1 + \mathbf{x}_i\beta_1 \\ \mathrm{link}\Big(\pi_{i2}/(1-\gamma_{i1})\Big) =& \kappa_2 + \mathbf{x}_i\beta_2 \\ \mathrm{link}\Big(\pi_{i3}/(1-\gamma_{i2})\Big) =& \kappa_3 + \mathbf{x}_i\beta_3 \end{aligned}$$

where $\pi_{i4}/(1-\gamma_{i3}) = 1$ and the κ_js are the usual “baseline-effect” parameters. These models can be fitted using the binomial likelihood functions

$$\begin{aligned} L_{(1)} &\propto \prod_i \pi_{i1}^{y_{i1}}(1-\pi_{i1})^{y_{i2}+y_{i3}+y_{i4}} \\ L_{(2)} &\propto \prod_i \left(\frac{\pi_{i2}}{1-\pi_{i1}}\right)^{y_{i2}} \left(1-\frac{\pi_{i2}}{1-\pi_{i1}}\right)^{y_{i3}+y_{i4}} \\ L_{(3)} &\propto \prod_i \left(\frac{\pi_{i3}}{1-\pi_{i1}-\pi_{i2}}\right)^{y_{i3}} \left(1-\frac{\pi_{i3}}{1-\pi_{i1}-\pi_{i2}}\right)^{y_{i4}} \end{aligned}$$

respectively. The three models are equivalent to a single continuation-ratio model of the form

$$\mathrm{link}(\delta_{ij}) = \kappa_j - \mathbf{x}_i\beta_j \qquad j = 1,\ldots,J-1$$

for the ordinal responses Y_i. The corresponding multinomial likelihood function

$$L(\pi_{ij}; y_j) \propto \prod_i \pi_{i1}^{y_{i1}}\pi_{i2}^{y_{i2}}\pi_{i3}^{y_{i3}}\pi_{i4}^{y_{i4}}$$

is used to fit the continuation-ratio model. Note that this is in fact the product of the likelihood functions, hence fitting the continuation-ratio model by maximum likelihood can be achieved using a routine for binary regression.

The sign in the continuation-ratio model is reversed from those in the binary regression models so that the effect parameters are positive if they cause a shift to higher categories (and hence are qualitatively similar to those from a cumulative probability model). The new command `ocratio` fits a model of the form

$$\mathrm{link}(\delta_{ij}) = \kappa_j - \mathbf{x}_i\beta \qquad j = 1,\ldots,J-1$$

where the explanatory effects are assumed to be constant across the response categories. An option is provided to test this assumption of constancy of effects by inspecting the likelihood-ratio between the first and third models above. Three link functions are available with the command; logit, probit and complementary-log-log. The β parameters from the logit model can be interpreted as log odds ratios of continuation-ratio probabilities and an option is provided to display these as odds ratios.

A curiosity in the family of continuation-ratio models is the fact that the complementary-log-log link model is equivalent to the cumulative probability counterpart (as demonstrated in Läärä and Matthews 1985). While the explanatory effects in both models are equal, the κ_j parameters of the continuation-ratio complementary-log-log model are transformations of the cut-point parameters of the ordered complementary-log-log model. This model can be displayed in either form with `ocratio`.

Syntax

`ocratio` *depvar* [*varlist*] [*weight*] [`if` *exp*] [`in` *range*] [, `link(`*linkname*`)` `eform` `test`
`level(`*#*`)` `cumulative`]

`ocrpred` [*type*] *newvarlist* [, `xb` `prob`]

where *linkname* is one of `logit` | `probit` | `cloglog`.

`fweight`s are allowed with `ocratio` and provide a very useful way of dealing with data that are stored in a contingency table format (see the example below).

`ocratio` shares most features of estimation commands; see [U] **26 Estimation and post-estimation commands**. To obtain predictions from the model, however, `ocrpred` should be used rather than `predict`.

Options for use with ocratio

`link(`*linkname*`)` specifies the link function. The default is `link(logit)` unless `cumulative` is requested in which case it is `link(cloglog)`.

`eform` displays the exponentiated coefficients with corresponding standard errors and confidence intervals as described in [R] **maximize**. For the logit link, exponentiation results in odds ratios of continuation-ratio probabilities; for the cloglog link, exponentiation results in ratios of discrete hazard functions.

`test` requests the calculation of the likelihood-ratio test of whether the effects of the explanatory variables are constant across the categories of the response variable.

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

`cumulative` requests the fitting of the continuation-ratio cloglog model and the display of the cut-points from the equivalent ordered cloglog model.

Options for use with ocrpred

`xb` requests the linear predictor, and one name should be supplied in *newvarlist*. This is the default if no options are specified with `ocratio`.

`prob` requests the predicted probabilities for each response category. There should be J names supplied in *newvarlist* where J is the value of `$S_3` (i.e., the number of response categories) from the previous `ocratio` command. An easy way of doing this is by using ranges, e.g., `ocrpred p1-p8, prob`.

Example: Pneumoconiosis in miners

We consider the data presented by McCullagh and Nelder (1989, 179) on the relationship between exposure in years and the degree of pneumoconiosis in coalface workers.

We define a variable `period` containing the exposure period in years (and `logt` containing the log-transformed values of period), a variable `response` which takes values from 1 to 3 corresponding to the three degrees of pneumoconiosis and `count` which contains the observed counts for each combination of `period` and `response`. We follow McCullagh and Nelder in fitting a continuation-ratio logit model in terms of `logt` using the command

```
. ocratio response logt [fweight=count], link(logit)
```

with the resulting output

```
Continuation-ratio logit Estimates                      Number of obs =     453
                                                        chi2(1)       =   94.01
                                                        Prob > chi2   =  0.0000
Log Likelihood = -205.5741                              Pseudo R2     =  0.1861
------------------------------------------------------------------------------
response |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
    logt |   2.321359   .3268181      7.103   0.000       1.680808    2.961911
------------------------------------------------------------------------------
 _cut1   |   8.733798   1.128289           (Ancillary parameters)
 _cut2   |   8.051302   1.179676
------------------------------------------------------------------------------
```

The coefficient for the log-transformed period effect can be displayed as an odds ratio by redisplaying the output with

```
. ocratio, eform
```

which produces (ignoring the leading statistics and ancillary parameters)

(*output omitted*)

```
------------------------------------------------------------------------------
response | Odds ratio   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
    logt |   10.18952   3.330118      7.103   0.000       5.369891    19.33489
------------------------------------------------------------------------------
```

The χ^2 statistic at the top of the output summarizes the improvement in fit over a model which contains no explanatory effects. The pseudo-R^2statistic is described in the Stata manual under `maximize` and should be used as a rough guide only to describe how much of the variation in the outcome is described by the current model. The fitted probabilities for each category of the outcome can be obtained using

```
. ocrpred pi1-pi3, prob
```

and fitted categories could be obtained using combinations of these categories, e.g., choosing the fitted category as j' where $\pi_{j'} = \max(\pi_j), \quad j = 1, \ldots, J$, or using a scored category with $\sum_j j\pi_j/J$. The log-likelihood for the model can then be recomputed from these fitted probabilities.

```
. gen unwtll_i=log(pi1)*(response==1)+log(pi2)*(response==2)+log(pi3)*(response==3)
. gen ll_i=(unwtll_i)*count
. gen ll=sum(ll_i)
. di "Log-likelihood = " ll[_N]
```

Technical notes

The error "matsize too small" may be encountered when using the `test` option but is remedied by issuing the `set matsize` command, e.g., `set matsize 150`. Also, when `ocratio` fits a model, the original data vectors of length n are expanded to length $(J-1) \times n$, so if J and n are large this can take an appreciable amount of time and there is potential for storage problems.

Saved results

S_1 Log-likelihood ratio for test of effect constancy assumption
S_2 Test degrees of freedom
S_3 Number of distinct response categories (with nonzero counts)

Note that S_1 and S_2 are only defined if the `test` option is specified.

Discussion

While the cumulative probability models may be used to model any ordinal response, interpretation of the model parameters is most straightforward for those ordinal responses which can be thought of as crude representations of some underlying continuous unobserved variable. The models are an appealing choice when the response categories are not of interest in their own right but only in what they tell us about a latent variable. The continuation-ratio models fitted using `ocratio` can also be employed to model any ordinal response but their interpretation is most straightforward when the ordinal scale represents a series of binary outcomes. These models are most appealing when the specific categories of the response scale are of interest in their own right.

References

Läärä, E. and J. Matthews. 1985. The equivalence of two models for ordinal data. *Biometrika* 72: 206–207.

McCullagh, P. and J. A. Nelder. 1989. *Generalized Linear Models*. 2d ed. London: Chapman & Hall.

sg87 Windmeijer's goodness-of-fit test for logistic regression

Jeroen Weesie, Utrecht University, J.Weesie@fss.ruu.nl

A popular goodness-of-fit statistic for binary models such as the logistic regression model is Pearson's X^2, defined as

$$X^2 = \sum_{i=1}^{m} \frac{m_i(y_i - \widehat{\pi}_i)^2}{\widehat{\pi}_i(1-\widehat{\pi}_i)}$$

where i runs over the covariate patterns, m_i is the number of replications under i, y_i is the proportion of observed successes in i, and $\widehat{\pi}_i$ the estimated probability of success in i.

It is useful to distinguish two cases. In the first case, the number m_i of observations with the same covariate pattern (replications), and hence the same expected probability, is "large" for every i. This is typically the case if there are only a few and discrete covariates. Then, according to standard asymptotic theory, X^2 is approximately χ^2 distributed with $m-p$ degrees of freedom, where again m is the number of covariate patterns, and p is the number of (independently) fitted parameters. Hosmer and Lemeshow (1989) refer to this result as *m-asymptotics*. Stata supports this statistic in the command `lfit`.

In the second case, there are only rare observations with the same covariate pattern ("replications"). This is usually the case with continuous covariates or with many discrete covariates. In this case, it is not useful to aggregate within covariate pattern, and we write $m_i = 1$ for each i, and so m equals the number of observations n. Without providing a formal theory, some authors suggest approximating X^2 with a χ^2-distribution with $n-p$ degrees of freedom (cf. Hosmer and Lemeshow 1989 on *n-asymptotics*). Below, I will refer to this test as the Pearson test. Hosmer and Lemeshow provide a X^2-like test statistic in which observations are grouped on the predicted probabilities rather than on the covariates. Again, a distribution theory on this test statistic is not available. Based on extensive simulations, Hosmer and Lemeshow suggest comparing their test statistic with a χ^2 distribution with $m-2$ degrees of freedom, where m is the number of groups. Stata supports this statistic in `lfit, group(#)`.

Thus, both goodness-of-fit tests for logistic regression with continuous covariates that are currently supported by Stata lack a firm formal basis in statistical theory. It seems to be little known that McCullagh (1986) and Windmeijer (1990) have derived asymptotic theory for the X^2-statistic for discrete choice models such as the logistic regression model. This theory is directly applicable to cases with continuous covariates. In particular, these authors show that under suitable regularity conditions a normalization of X^2 is asymptotically *normally* distributed, and, hence, the square of the normalized statistic is $\chi^2(1)$-distributed. More specifically, it can be shown that

$$H_n = \frac{(X^2-n)^2}{n\widehat{\sigma}_n^2} \rightarrow \chi^2(1)$$

Here

$$\sigma_n^2 = \frac{1}{n}\sum_{i=1}^{n} \frac{1-2\pi_i}{\pi_i(1-\pi_i)} - v_n \Omega_n^{-1} v_n$$

where

$$v_n = \frac{1}{n}\sum_{i=1}^{n}(1-2\pi_i)x_i$$

with x_i the covariate vector for i's observation, and

$$\Omega_n = \frac{1}{n}\sum_{i=1}^{n} \pi_i(1-\pi_i)x_i x_i'$$

is the mean Fisher-information. Using this asymptotic result, one could construct a goodness-of-fit test that rejects a logistic regression model if H_n is large, say $H_n > 3.84$ (at $\alpha = 0.05$). Rejecting the model by itself does not, of course, suggest improved models. For instance, important variables may be missing, the logit link may not be appropriate, etc.

A number of remarks on this asymptotic result and the associated test are in order. First, σ_n^2 is in general not equal to 2, as would be expected if the X^2 itself was approximately χ^2 distributed. Thus, in general, X^2 is *not* χ^2 distributed. Examples are easily constructed in which the χ^2 approximation is arbitrarily inaccurate (See Windmeijer 1990 for an example with $\sigma_n^2 \rightarrow 0.034$). Second, the approximating $\chi^2(1)$ distribution of H_n does not depend directly on the number of observations and the number of estimated parameters. It comes to mind to compare the result of McCullagh and Windmeijer to the approximation of a χ^2-distributed variate Y_k with a normal variate

$$\frac{Y_k - k}{\sqrt{2k}} \approx N(0,1)$$

which is asymptotically true for large degrees of freedom k, but not accurate for small k (See Hoaglin 1977 for details and improved normal approximations of the χ^2 distribution). This suggests that the asymptotic $\chi^2(1)$ distribution of H_n may be reasonably accurate for very large numbers of observations only. Third, v_n is degenerate for $\pi = 0$ or $\pi = 1$. Windmeijer (1990) suggests excluding observations with extreme (estimated) probabilities in computing H_n (see the option `eps` of my command `lfitx2`).

Windmeijer (1990) also derives the normalizing constants of Pearson's X^2 statistic for other binary regression models such as probit-regression (cf. `probit` in Stata) and complementary-log-regression. Windmeijer (1994) generalizes these results to a multinomial logistic regression model (cf. `mlogit` in Stata) and to conditional logistic regression (cf. `clogit` in Stata). Currently, Stata does not provide goodness-of-fit statistics for the polytomous case. I hope to provide implementations of Windmeijer's goodness-of-fit tests in the near future.

I wrote a command `lfitx2` that computes Windmeijer's goodness-of-fit test based on the normalization H_n of Pearson's X^2. Like `lfit`, it can be called only after `logistic`. It will automatically use the same sample as `logistic`. Note that currently `lfitx2` does not support weights or out-of-sample usage.

Syntax

`lfitx2` [, `eps(`*#*`)`]

Option

`eps(`*#*`)` specifies that all observations with expected probability $p <$`eps` or $p > 1-$`eps` are excluded from computing the X^2 statistic. The default value of `eps` is `1e-5`.

Example 1

As a first example, I consider the same data that are used to illustrate the `lfit` command in the Stata Reference Manual. These data come from a study of the risk factors associated with low birth weight. After estimating a logistic regression model with eight covariates, one can obtain three goodness-of-fit tests, the first two of which are copied literally from the manual.

```
. lfit

Logistic model for low, goodness-of-fit test

       number of observations =       189
 number of covariate patterns =       182
             Pearson chi2(173) =      179.24
                   Prob > chi2 =        0.3567

. lfit, group(10)

Logistic model for low, goodness-of-fit test
(Table collapsed on quantiles of estimated probabilities)

       number of observations =       189
             number of groups =        10
      Hosmer-Lemeshow chi2(8) =         9.65
                   Prob > chi2 =        0.2904

. lfitx2

Logistic model for low, Pearson´s goodness-of-fit test

         number of observations =       189
                   Pearson´s X2 =   182.023
               Prob > chi2(180) =    0.4438        Beware: Invalid!
Windmeijer´s H = normalized(X2) =     0.692
                 Prob > chi2(1) =    0.4054
```

We conclude that neither of the three goodness-of-fit tests rejects the model. Note that in the output of `lfitx2`, we also include the invalid approximate p value, based on a $\chi^2(n-p)$ distribution, including a warning message that it will generally be too inaccurate to be of any practical use.

Example 2

Both as a second illustration of the command `lfitx2` and as a step towards the analysis of the conditions under which different goodness-of-fit statistics are appropriate, I conducted some simple Monte Carlo simulation studies that

1. study the distribution of Pearson's X^2 and Windmeijer's H (as produced by `lfitx2`) for different sample sizes, and
2. compare the goodness-of-fit tests due to Pearson, Hosmer and Lemeshow, and Windmeijer.

In the first of these experiments, synthetic datasets are constructed by sampling from

$$z \sim N\left(-\frac{1}{2}, 1\right)$$

$$y|z \sim \text{Bin}\left(1, \frac{\exp(1+z)}{1+\exp(1+z)}\right)$$

For each of these simulated datasets, the true logistic regression model with covariate z and the constant is estimated, and Pearson's X^2 and Windmeijer's H are computed. Next to moments and tail probabilities of the distributions of X^2 and H, we also tested whether H is $\chi^2(1)$-distributed using an Anderson–Darling test (Royston 1996). In the next table, we report some results of this study based on 1000 replications for each of the sample sizes $n = 40$, $n = 100$, and $n = 1000$.

	Pearson's X^2		Windmeijer's H		Proportion H_0 rejected			Anderson–Darling	
n	mean	variance	mean	variance	$\alpha = .10$	$\alpha = .05$	$\alpha = .01$	A2	p
40	39.4	16.6	.86	6.60	.059	.049	.027	73.0	.000
100	99.5	41.0	1.04	4.93	.076	.042	.014	2.2	.069
1000	999.8	130.0	1.01	2.25	.095	.048	.010	0.4	.820

If X^2 were approximately χ^2 distributed, the variance of X^2 would be approximately twice the mean of X^2. From the second and third column of the table it is obvious that the χ^2 distribution is indeed a very poor approximation of X^2. From the other columns of the table, we can conclude that the $\chi^2(1)$ distribution provides a reasonable approximation for the distribution of Windmeijer's H for $n = 100$ and, in particular, for $n = 1000$; the variance of H is approximately twice the mean of H; the tail-probabilities of H are reasonably close to the nominal levels, and Anderson–Darling's test rejects H_0 only barely ($n = 100$) or not at all ($n = 1000$). Note though that for small sample size (here $n = 40$), the $\chi^2(1)$ of H is very poor indeed.

In our second experiment, we elaborate on one of the examples of Windmeijer (1990). Here synthetic datasets are constructed that consist of observations that are generated according to the logistic regression specification

$$z \sim N\left(-\frac{1}{2}, 1\right)$$

$$\text{LP} = z + \gamma \ln(|z|)$$

$$y \sim \text{Bin}\left(1, \frac{\exp(\text{LP})}{1+\exp(\text{LP})}\right)$$

On a synthetic dataset, I estimated a logistic regression model with z as the only covariate in addition to the constant. Consequently, if $\gamma \neq 0$, the estimated model is misspecified due to the *omitted variable* $\ln(|z|)$. γ measures the extent to which the estimated model is misspecified. For various values of γ, 1000 datasets were generated, the logistic regression model was estimated, and the Pearson, the Hosmer–Lemeshow and the Windmeijer tests were computed for $\alpha = .10$, $\alpha = .05$, and $\alpha = .01$. Above, I discussed the suspicion that Windmeijer's asymptotic result may be accurate only for large number of observations. Thus, in the simulation I varied the sample size between 40 and 1000. Finally, since all inference in the logistic regression model is conditional on the covariates, as in the first example, the z covariates were generated only once for each sample size.

In Weesie (1998), I published an extension, `simul2`, of `simul` that facilitates conducting full-factorial parameterized Monte Carlo simulations. This example also serves as an illustration of these extensions. With `simul2`, one should write a simulation program that depends on the scalars `gamma` (true coefficient of $\ln(|z|)$) and the sample size `Nobs`, and issue a simulation command with the additional argument `par(gamma 0-1/0.25, Nobs 100 1000)`. As an illustration of `simul2`, the main part of the code for running the simulation is listed below. (The actual code is slightly more complicated because it has to deal adequately with the case in which z perfectly predicts y, where `logistic` produces nonstandard output.)

```
* simulation example Windmeijer 1990, p73 (using -simul2-)
* parameters
*   gamma measures the size of the omitted variable  levels: 0-1/0.25
*   Nobs is the sample size                          levels: 40 200 1000

program define logsim
        if "`1´" == "?"
                * setup information for driver -simul2-
                global S_1 "gamma Nobs P pP HL pHL W pW"
                exit
        * synthetic data according to Windmeijer´s model 2
          gen lp = 1 + z + gamma*ln(abs(z))
          gen y = uniform() > 1/(1+exp(-lp))
        * fit logistic regression model to cases 1..Nobs
          logistic y z  if _n <= Nobs
        * Pearson´s test
          lfit
          scalar pP = chiprob(Nobs-2,$S_3)
        * Hosmer-Lemeshow statistic
          lfit, gr(10)
          scalar pHL = chiprob($S_4,$S_3)
        * Windmeijer´s statistic
          lfitx2
          scalar pW  = chiprob(1,$S_3)
        * post results for driver program
          post `1´ gamma Nobs P pP HL W pHL pW
          drop lp y
end
```

In our simulation code, we set the `seed` of the random generator to ensure that results will be reproducible, and generate a common variate z—all inference is conditional on the covariates.

```
set seed 1432
drop _all
set obs 1000
gen z = -0.5 + invnorm(uniform())
```

We are now ready to run the simulation with `gamma` running between 0 and 1 in steps of .25, and `Nobs` running over the levels 40, 100, and 1000.

```
. simul2 logsim, rep(1000) par(gamma 0-1/0.25, Nobs 40 100 1000) dots report
```

The command `simul2` summarizes its main inputs and displays progress markers as the simulation runs.

```
Simulation program: logsim
Parameter  #values  values
gamma         5     0 0.25 0.50 0.75 1
Nobs          3     40 100 1000
number of conditions:       15
replications per condition: 1000
[0,40] ......................   (output omitted)
[0.25,40] ...................   (output omitted)
  (output omitted)
[1,1000] ....................   (output omitted)
```

To estimate the power of the tests, we compute proportions of rejections at different significance levels:

```
* compute proportions of rejections for alpha = 0.01, 0.05, and 0.10
for 01 05 10, l(any): gen prP@  = pP  < 0.@
for 01 05 10, l(any): gen prHL@ = pHL < 0.@
for 01 05 10, l(any): gen prW@  = pW  < 0.@
```

The table below describes the proportions of simulations in which the respective tests rejected the logistic regression model at .10, .05, and .01 significance levels, based on 1000 replications for each of the `gamma, Nobs` conditions.

		Pearson			Hosmer–Lemeshow			Windmeijer		
n	γ	.100	.050	.010	.100	.050	.010	.100	.050	.010
40	0	.034	.017	.012	.119	.050	.012	.067	.047	.029
	0.25	.010	.003	.002	.112	.056	.005	.042	.024	.009
	0.50	.004	.000	.000	.136	.068	.009	.062	.025	.005
	0.75	.005	.003	.002	.228	.113	.018	.103	.051	.012
	1.00	.006	.004	.002	.310	.169	.025	.189	.113	.024
100	0	.011	.007	.003	.077	.042	.008	.030	.020	.012
	0.25	.012	.009	.004	.083	.033	.006	.039	.028	.011
	0.50	.015	.008	.003	.071	.036	.005	.029	.020	.008
	0.75	.025	.014	.008	.074	.038	.010	.036	.027	.018
	1.00	.027	.019	.008	.092	.043	.006	.030	.022	.013
1000	0	.000	.000	.000	.096	.050	.008	.109	.064	.022
	0.25	.000	.000	.000	.578	.457	.217	.597	.457	.205
	0.50	.000	.000	.000	.998	.995	.965	.966	.931	.819
	0.75	.000	.000	.000	1.000	1.000	1.000	.949	.927	.864
	1.00	.000	.000	.000	1.000	1.000	1.000	.712	.655	.535

Again, we see that the Pearson test performs very badly. The difference between the Hosmer–Lemeshow and Windmeijer tests are rather small. Windmeijer's test seems to be less biased than the test by Hosmer–Lemeshow, e.g., the probability of invalid rejections of H_0 is closer to the significance level α for Windmeijer's test than for the Hosmer–Lemeshow test. On the other hand, the Hosmer–Lemeshow test seems to be somewhat more powerful in detecting omitted variables. Note that the power of the Windmeijer test seems to be nonmonotonic in the size of the omitted variable. I encountered this curious fact also in a number of other simulations that are not reported here. Finally, at the relatively small sample sizes 40 and 100, all tests had substantial biases and little power.

It is tempting to interpret this table as "Hosmer–Lemeshow and Windmeijer's test are roughly equivalent," and, hence, we can restrict to using one of these goodness-of-fit tests. This is of course not valid since the tests are not as highly correlated as might be expected from their common relation to the Pearson test. Consider as an example, the condition `gamma`=.25 and `Nobs`=1000. In our 1000 replications, we found the following cross-classification of test results:

		Windmeijer accept H_0	 reject H_0
Hosmer–Lemeshow	accept H_0	290	167
	reject H_0	167	376

Thus, the tests by Hosmer–Lemeshow and by Windmeijer are rank-correlated at a modest level of .32. Consequently, it makes good sense to use both tests.

Example 3

In the third simulation, I compare the power of the three goodness-of-fit tests against a *misspecification of the link function*. In addition, I also include Pregibon's link test (in Stata's `linktest` command) since that may be expected to be more powerful against this particular type of model misspecification. I analyzed the stochastic model

$$\begin{aligned} x &\sim N(0,1) \\ y|x &\sim f(x) \end{aligned}$$

for the three link functions

$$\text{logit} : f_1(z) = \frac{\exp(z)}{1+\exp(z)}$$

$$\text{probit} : f_2(z) = \Phi(z/1.6)$$

$$\text{complementary} - \log : f_3(z) = \exp(-\exp(-z + \ln(\ln 2)))$$

Note that the probit-link is scaled by a constant 1.6 to make the probit link as much as possible comparable to the logit link. The complementary-log link was shifted by $\ln(\ln 2)$ in order to make the marginal distributions of y under the three link-specifications the same (namely, the unconditional probability $\Pr(y=1) = .5$ in each case).

In the simulation, we again varied sample size between 40, 100, and 1000. For each synthetic dataset, a logistic regression model with covariates 1 and x was fit, and the three goodness-of-fit tests and the link test were computed at the significance levels .10, .05, and .01. In the next table, proportions of rejections among 1000 replications are reported.

		Pearson			Hosmer–Lemeshow			Windmeijer			link test		
n	link	.10	.05	.01	.10	.05	.01	.10	.05	.01	.10	.05	.01
	logit	.030	.018	.012	.090	.043	.006	.060	.046	.027	.090	.047	.001
40	probit	.027	.014	.003	.089	.033	.004	.031	.019	.009	.068	.023	.001
	comp-log	.086	.056	.025	.105	.057	.013	.051	.036	.025	.090	.042	.003
	logit	.005	.003	.003	.113	.042	.003	.045	.028	.016	.069	.025	.005
100	probit	.006	.004	.001	.109	.050	.013	.031	.020	.007	.051	.019	.001
	comp-log	.019	.017	.012	.082	.039	.010	.000	.000	.000	.043	.012	.001
	logit	.001	.000	.000	.077	.041	.007	.085	.041	.011	.085	.040	.012
1000	probit	.000	.000	.000	.107	.052	.012	.102	.041	.004	.091	.045	.006
	comp-log	.097	.051	.013	.275	.182	.062	.170	.118	.058	.560	.415	.195

The conclusions on the tests by Pearson, Hosmer–Lemeshow, and Windmeijer are comparable to the previous experiment: Pearson performs poorly. Windmeijer's test has less bias than Hosmer–Lemeshow, but Hosmer–Lemeshow is more powerful. The logit link and the probit link cannot be distinguished by any of the tests. The dedicated link test is more powerful than the omnibus goodness-of-fit tests for the complementary-log misspecification of the link function.

References

Hoaglin, D. 1977. Exploring a table of percentage points of χ^2. *Journal of the American Statistical Association* 72: 508–515.

Hosmer, D. W. and S. Lemeshow. 1989. *Applied Logistic Regression*. New York: John Wiley & Sons.

McCullagh, P. 1986. The conditional distribution of goodness-of-fit statistics for discrete data. *Journal of the American Statistical Association* 81: 104–107.

Royston, P. 1996. sg47: A plot and test for the chi-square distribution. *Stata Technical Bulletin* 29: 26–27. Reprinted in *Stata Technical Bulletin Reprints* vol. 5, pp. 142–144.

Weesie, J. 1998. ip25: Parameterized Monte Carlo simulations: An enhancement to the simulation command. *Stata Technical Bulletin* 43: 13–15.

Windmeijer, F. A. G. 1990. The asymptotic distribution of the sum of weighted squared residuals in binary choice models. *Statistica Neerlandica* 44(2): 69–78.

——. 1994. A goodness-of-fit test in the multinomial logit model based on weighted squared residuals. *Statistica Neerlandica* 48(3): 271–283.

sg88 Estimating generalized ordered logit models

Vincent Kang Fu, UCLA, vfu@pop.ben2.ucla.edu

Description

The command `gologit` estimates a regression model for ordinal variables that I call the generalized ordered logit model. Stata's `ologit` command also estimates a regression model for ordinal variables, but the model that `ologit` estimates imposes what is called the proportional odds assumption on the data. The generalized ordered logit model relaxes this assumption. The proportional odds assumption is that explanatory variables have the same effect on the odds that the dependent variable is above any dividing point. `gologit` relaxes the proportional odds assumption by allowing the effects of the explanatory variables to vary with the point where the categories of the dependent variable are dichotomized.

Researchers have given the generalized ordered logit model brief attention (e.g., Agresti 1984, Brant 1990, Clogg and Shihadeh 1994, McCullagh and Nelder 1989, Maddala 1983), but usually pass over it in favor of the more restrictive proportional odds model, even though it is not clear that the proportional odds assumption generally holds (Long 1997). The model `gologit` estimates is very similar to a model proposed by Peterson and Harrell (1990), although they conceptualize it slightly differently.

Syntax

`gologit` *depvar varlist* [*weight*] [`if` *exp*] [`in` *range*]
[`,` `cluster(`*varname*`)` `level(`*#*`)` `or` `robust`]

This command shares the features of all estimation commands. `gologit` typed without arguments redisplays previous results. The options `level(`*#*`)` and `or` may be given when redisplaying results.

`aweight`s, `fweight`s, and `pweight`s are allowed. Using `pweight`s or `cluster()` implies the `robust` option.

Coefficients are estimated using Stata's `ml` interface. Stata's `ologit` command is used to produce starting values. The actual values taken on by the dependent variable are irrelevant except that larger values are assumed to correspond to "higher" outcomes.

Options

`cluster(`*varname*`)` specifies that the observations are independent across groups (clusters) but not necessarily within groups. *varname* specifies to which group each observation belongs; e.g., `cluster(`*personid*`)` in data with repeated observations on individuals. See [U] **26.10 Obtaining robust variance estimates**. `cluster` can be used with `pweight`s to produce estimates for unstratified cluster-sampled data. Specifying `cluster` implies `robust`.

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

`or` reports the estimated coefficients transformed to odds ratios, i.e., exp(b) rather than b. Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated. `or` may be specified at estimation or when redisplaying previously estimated results.

`robust` specifies the Huber–White sandwich estimator of variance is to be used in place of the traditional calculation; see [U] **26.10 Obtaining robust variance estimates**. `robust` combined with `cluster` allows observations which are not independent within cluster (although they may be independent between clusters).

Remarks

More formally, suppose we have an ordinal dependent variable Y which takes on the values $0, 1, 2, \ldots, m$. The generalized ordered logit model estimates a set of coefficients and a constant for each of the $m-1$ points at which the dependent variable can be dichotomized. These sets of coefficients B_k correspond to a set of cumulative distribution functions:

$$P(Y < k) = F(-XB_k) \qquad k = 1, \ldots, m$$

From this set of cumulative distribution functions, it is straightforward to derive formulas for the probabilities that Y will take on each of the values $0, 1, \ldots, m$:

$$\begin{aligned} P(Y=0) &= F(-XB_1) \\ P(Y=1) &= F(-XB_2) - F(-XB_1) \\ P(Y=j) &= F(-XB_{j+1}) - F(-XB_j) \qquad j = 1, \ldots, m-1 \\ P(Y=m) &= 1 - F(-XB_m) \end{aligned}$$

The generalized ordered logit model uses the logistic distribution as the cumulative distribution, although other distributions are certainly possible. The logistic distribution allows researchers to interpret this model in terms of logits:

$$\log[P(Y \geq k)/P(Y < k)] = XB_k \qquad k = 1, \ldots, m$$

The proportional odds property of Stata's `ologit` command restricts the B_k coefficients to be the same for every $k = 1, \ldots, m$.

Note that unlike models such as OLS regression and binary logit, the generalized ordered logit model imposes explicit restrictions on the range of the X variables. Since probabilities are by definition constrained to be between 0 and 1, valid combinations of the X variables must satisfy the following inequalities:

$$\begin{aligned} -XB_2 &\geq -XB_1 \\ -XB_3 &\geq -XB_2 \\ &\cdots \\ -XB_m &\geq -XB_{m-1} \end{aligned}$$

Example

Consider the familiar auto data. These data contain information on 1978 repair records of automobiles. Here is a table of the data:

```
. tab rep78
Repair      |
Record 1978 |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |          2        2.90        2.90
          2 |          8       11.59       14.49
          3 |         30       43.48       57.97
          4 |         18       26.09       84.06
          5 |         11       15.94      100.00
------------+-----------------------------------
      Total |         69      100.00
```

Since small cell sizes are a big problem for `gologit`, let us combine the two lowest categories (poor, fair) with the middle category (average). The new variable has three categories: average or worse, good, and excellent. Here is a table of the data we will be modeling with each category given a descriptive label:

```
. tab rep78
Repair      |
Record 1978 |      Freq.     Percent        Cum.
------------+-----------------------------------
   avg/wors |         40       57.97       57.97
       good |         18       26.09       84.06
   excellen |         11       15.94      100.00
------------+-----------------------------------
      Total |         69      100.00
```

Assume that we wish to know if repair records are related to where the car was manufactured (foreign or domestic), its length (as a proxy for size), and its engine displacement. Using gologit, we would estimate the following model:

```
. gologit rep78 foreign length displ

Iteration 0:  Log Likelihood = -66.194631
Iteration 1:  Log Likelihood = -50.481626
Iteration 2:  Log Likelihood = -47.270644
Iteration 3:  Log Likelihood = -46.002971
Iteration 4:  Log Likelihood = -45.871464
Iteration 5:  Log Likelihood =   -45.8704
Iteration 6:  Log Likelihood =   -45.8704

Generalized Ordered Logit Estimates                 Number of obs    =      69
                                                    Model chi2(6)    =   40.65
                                                    Prob > chi2      =  0.0000
Log Likelihood =    -45.8704000                     Pseudo R2        =  0.3070

------------------------------------------------------------------------------
   rep78 |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
mleq1    |
 foreign |   3.659307   1.029914      3.553   0.000       1.640713    5.677901
  length |   .0372637   .0278299      1.339   0.181      -.0172819    .0918093
   displ |  -.0068156   .0069198     -0.985   0.325       -.020378    .0067469
   _cons |  -7.093281   4.459775     -1.591   0.112      -15.83428    1.647717
---------+--------------------------------------------------------------------
mleq2    |
 foreign |   .1758222   1.162968      0.151   0.880      -2.103553    2.455197
  length |   .1879727   .0678996      2.768   0.006        .054892    .3210535
   displ |   -.094616   .0315209     -3.002   0.003      -.1563959   -.0328362
   _cons |  -21.75103   7.993136     -2.721   0.007      -37.41729    -6.08477
------------------------------------------------------------------------------
```

The coefficients under the heading mleq1 correspond to the first dividing point, between average or worse and good. The coefficients under the heading mleq2 correspond to the second dividing point, between good and excellent.

In essence, the coefficients correspond to the two possible cumulative binary logits that can be formed from this three-category dependent variable. The coefficients in mleq1 correspond to the logit formed from the two categories (good, excellent) and (average or worse). The coefficients in mleq2 correspond to the logit formed from the two categories (excellent) and (good, average or worse). If the dependent variable had more categories, gologit would produce additional panels of coefficients corresponding to the additional dividing points.

The estimates indicate that foreign cars are significantly more likely than domestic cars to have 1978 repair records that are good or excellent instead of average or worse. However, foreign and domestic cars do not differ in the likelihood of having excellent instead of good or worse repair records.

Longer cars are more likely to have excellent repair records than to have good or worse repair records, although length has no effect on the likelihood of a car having an average or worse versus a good or better repair record.

Cars with larger engine displacements are more likely to have good or worse repair records than to have excellent repair records. However, engine displacement has no effect on the likelihood of being above or below the other dividing point, between average or worse and good.

These results indicate that the relationship between repair record and the explanatory variables is quite complex. None of the explanatory variables has a consistent, uniform effect on repair record. Thus, it would be inappropriate to conclude that foreign cars have better repair records, although these results would justify the conclusion that foreign cars are more likely than domestic cars to have good or excellent instead of average or worse repair records.

Since the proportional odds model estimated by ologit is a bit easier to interpret, we may wish to see if these data violate the proportional odds assumption. To do that, we can perform a Wald test of whether the coefficients under the heading mleq1 are equal to the coefficients under the heading mleq2.

```
. test [mleq1=mleq2]

 ( 1)  [mleq1]foreign - [mleq2]foreign = 0.0
 ( 2)  [mleq1]length - [mleq2]length = 0.0
 ( 3)  [mleq1]displ - [mleq2]displ = 0.0

           chi2(  3) =    8.32
         Prob > chi2 =    0.0399
```

The chi-square statistic is significant at conventional levels, suggesting that these data do violate the proportional odds assumption. Consequently, the model estimated by ologit is inappropriate for these data.

References

Agresti, A. 1984. *Analysis of Ordinal Categorical Data*. New York: John Wiley & Sons.

Brant, R. 1990. Assessing proportionality in the proportional odds model for ordinal logistic regression. *Biometrics* 46: 1171–1178.

Clogg, C. and E. Shihadeh. 1994. *Statistical Models for Ordinal Variables*. Thousand Oaks, CA: Sage Publications.

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

McCullagh, P. and J. Nelder. 1989. *Generalized Linear Models*. 2d ed. New York: Chapman & Hall.

Maddala, G. 1983. Limited-dependent and qualitative variables in econometrics. *Econometric Society Monographs no. 3*. New York: Cambridge University Press.

Peterson, B. and F. Harrell, Jr. 1990. Partial proportional odds models for ordinal response variables. *Applied Statistics* 39: 205–217.

sg89 Adjusted predictions and probabilities after estimation

Kenneth Higbee, Stata Corporation, khigbee@stata.com

Syntax

adjust *var* [= *#*] [*var* [= *#*] ...] [if *exp*] [in *range*] , by(*varlist*) [{xb|pr} {stdp|stdf|noerror}

generate(*newvar1* [*newvar2*]) xblabel([“]*text*[”]) prlabel([“]*text*[”]) stdplabel([“]*text*[”])

stdflabel([“]*text*[”]) *tabdisp_options*]

where the allowed tabdisp options are: format(%*fmt*), center, left, cellwidth(*#*), csepwidth(*#*), scsepwidth(*#*), and stubwidth(*#*). For details, see the manual or on-line help for tabdisp.

Description

After an estimation command, adjust provides adjusted predictions of $x\beta$ (the means in a linear-regression setting) or probabilities (available after certain estimation commands). The estimate is computed for each level of the by variable(s) setting the variable(s) specified in *var* [= *#*] [*var* [= *#*] ...] to their mean or to the specified number if the = # part is used. Variables used in the estimation command but not included in either the by variable list or the adjust variable list are left at their current values, observation by observation. In this case, adjust displays the average estimated prediction or probability for each level of the by variables.

Options

by(*varlist*) is required and specifies the variable(s) whose levels determine the subsets of the data for which adjusted predictions are to be computed. The variables in the by option are not required to be involved in the original estimation command.

xb indicates that the linear prediction from the estimation command is to be used. This produces predicted values (means in the linear-regression setting) and is equivalent to the xb option of predict. Realize that depending on the estimation command, the xb values may not be in the original units of the dependent variable.

pr is an alternative to xb and indicates that predicted probabilities are to be computed. The pr option is only allowed after a few commands (blogit, bprobit, dprobit, logistic, logit, and probit). The default is xb if pr is not specified.

stdp, stdf, and noerror specify the kind of standard error reported for each mean. stdp, the default, uses the standard error of the linear prediction and is available only with the xb option. stdf uses the standard error of the forecast which is the standard error of the linear prediction plus the residual. stdf may be specified only with the xb option and after linear-regression estimation commands (boxcox, corc, fit, hlu, prais, regdw, regress, and rreg). noerror specifies that no error term is to be used and hence no standard errors for the means reported in the table.

generate(*newvar1* [*newvar2*]) generates one or two new variables. If one variable is specified, then the adjusted predictions for each observation are generated in *newvar1* (holding the appropriate variables to their means or other specified values). If pr is specified, then the predictions are probabilities. If *newvar2* is specified, then the standard errors from either the stdp or stdf option are placed in the second variable.

xblabel(“*text*”), prlabel(“*text*”), stdplabel(“*text*”), and stdflabel(“*text*”) allow you to change the labels for xb, pr, stdp, and stdf in the output produced by adjust. This also changes the variable labels for the variables created by the generate option.

Remarks

If you have restricted your estimation command to a portion of the data using `if` or `in`, then you will generally want to use the same conditions with `adjust`. However, there may be legitimate reasons for using different data to perform the estimation and to obtain adjusted predictions.

An interesting way to use `adjust` is after using `xi` with the estimation command. Another interesting use of `adjust` is to specify a categorical variable both in the variable list being set to a specific value and as a `by` variable. This is helpful in examining the predictions for several groups as if they were set at a particular group's value (possibly also holding some other variables to certain values also).

`adjust` is implemented to follow the same estimation commands as listed in the help for `predict`. However, `adjust` currently is not allowed after `anova` (because we can not obtain the variable names from the beta vector), `areg` (because `predict` doesn't handle the absorbed variable as expected), and multiple-equation estimation commands. `adjust` only uses `predict`. It does not use any specialized predict commands such as `fpredict`, `glmpred`, and `xtpred`.

`adjust` is similar to `adjmean` and `adjprop` whose latest versions are illustrated in STB-43 (Garrett 1998). There are some differences between this command and those. First, `adjmean` and `adjprop` internally use `regress` and `logistic` and are not available for other estimation commands. In comparison, `adjust` does not perform the estimation command, but instead is a post estimation command and will work after a large number of single equation estimation commands. A second difference is that `adjust` lets you specify up to seven variables in the `by` option. `adjust` also allows `by` variables that were not used in the estimation command. Variables used in the estimation command are left at their individual values if the variables are not specified in the `adjust` command. Another difference is that `adjmean` and `adjprop` compute confidence intervals and optionally display them in a graph, while `adjust` does not. `adjust` really is a front end process for `predict`. It sets up the values at which predictions are desired and then displays them using `tabdisp`. This means that `adjust` does not provide confidence intervals directly, but will provide `stdp` or `stdf` estimates of error under the same conditions that `predict` would produce them.

The adjusted predictions are displayed using the `tabdisp` command. In addition to the `adjust` command options that allow you to control the labeling of the prediction and error, several options from `tabdisp` can be specified for control over the final appearance of the table. These output control options are not used in the examples that follow.

Examples

The ubiquitous Stata auto dataset is used in the following examples to illustrate the `adjust` command.

Example 1

Let's pretend that we want to understand automobile price as a function of whether the car was manufactured domestically. We might first look at the average price for domestic and foreign cars.

```
. table foreign, c(mean price sd price)

----------+-----------------------
 Car type | mean(price)   sd(price)
----------+-----------------------
 Domestic |    6072.423    3097.104
  Foreign |    6384.682    2621.915
----------+-----------------------
```

These average prices do not seem very far apart. However, we really don't think that these averages are telling us the full story since there are other variables that have impact on the price of automobiles. As an (admittedly fake) first attempt at modeling automobile price we decide to perform a regression.

```
. regress price mpg weight turn foreign
  Source |       SS       df       MS                  Number of obs =      74
---------+------------------------------               F(  4,    69) =   19.23
   Model |  334771309     4  83692827.3                Prob > F      =  0.0000
Residual |  300294087    69  4352088.22                R-squared     =  0.5271
---------+------------------------------               Adj R-squared =  0.4997
   Total |  635065396    73  8699525.97                Root MSE      =  2086.2

------------------------------------------------------------------------------
   price |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
     mpg |  -.4660076   73.51407     -0.006   0.995      -147.1226    146.1905
  weight |   4.284532   .7404967      5.786   0.000       2.807282    5.761782
    turn |  -229.2059   114.2423     -2.006   0.049      -457.1131   -1.298687
 foreign |   3221.415   706.4847      4.560   0.000       1812.017    4630.813
   _cons |   1368.197   4887.597      0.280   0.780      -8382.291    11118.68
------------------------------------------------------------------------------
```

We now ask ourselves what the price of domestic and foreign cars would be if they were compared at identical values of the variables: `turn`, `weight`, and `mpg` under this regression model. We decide to compare them with these three variables set at their mean value. The `adjust` command provides us with the answer.

```
. adjust mpg weight turn, by(foreign)
-------------------------------------------------------------------------------
Covariates set to mean: mpg = 21.297297, weight = 3019.4595, turn = 39.648647
-------------------------------------------------------------------------------

----------+-----------------------
 Car type |         xb        S.E.
----------+-----------------------
 Domestic |   5207.539    320.8226
  Foreign |   8428.954    552.5154
----------+-----------------------
```

We see that under this regression model foreign cars are predicted to be more expensive than domestic cars when compared at the average value of `mpg`, `weight`, and `turn`. The prediction standard errors are also presented in the output. What if instead we wanted to know the forecast standard errors?

```
. adjust mpg weight turn, by(foreign) stdf
-------------------------------------------------------------------------------
Covariates set to mean: mpg = 21.297297, weight = 3019.4595, turn = 39.648647
-------------------------------------------------------------------------------

----------+-----------------------
 Car type |         xb     S.E.(f)
----------+-----------------------
 Domestic |   5207.539    2110.691
  Foreign |   8428.954    2158.092
----------+-----------------------
```

Understand the difference between these two tables. The reported means are the same but their reported standard errors differ. The first table uses `stdp`, the standard error of the linear prediction. This standard error measures our uncertainty as to the mean and that uncertainty is caused by our uncertainty of the estimated model coefficients. Were our model estimated on an infinite population, we would be certain as to the model's coefficients and hence the mean of each population. These standard errors would then be zero.

Even if we knew the mean with certainty—even if the `stdp` standard errors were zero—we would be uncertain as to the price of each individual car. This is because individual cars have individual prices drawn from a distribution whose mean we would know; $\text{price}_j = \mu_j + \epsilon_j$.

In the second table the reported standard errors are noticeably larger than those in the first. These standard errors reflect our total uncertainty as to the price of individual cars, $\sqrt{\sigma_\mu^2 + \sigma_\epsilon^2}$. This uncertainty is based on our uncertainty as to the mean itself, σ_μ—the prediction's standard error—and the inherent uncertainty because of the unmeasured characteristics of the individual cars themselves, σ_ϵ, the residual standard error. These two components, appropriately combined, are called the forecast standard error.

We might also want to know the predicted cost for domestic and foreign cars when some of the variables are set at particular values. For instance if we want to compare the predicted price when `mpg` is 25, `turn` is 35.2, and `weight` is at its mean value, we would do the following:

```
. adjust mpg=25 weight turn=35.2, by(foreign) noerror
-------------------------------------------------------------------------------
Covariate set to mean: weight = 3019.4595
Covariates set to value: mpg = 25, turn = 35.2
-------------------------------------------------------------------------------

----------+-----------
 Car type |         xb
----------+-----------
 Domestic |    6225.47
  Foreign |   9446.885
----------+-----------
```

We include the `noerror` option simply to illustrate that including it suppresses any standard error calculation. In practice, we would have specified `stdp` or `stdf`.

Let's imagine that we now want to find out the predicted average price of foreign and domestic cars under this regression model when `mpg` and `weight` are set to their overall means but the `turn` variable is left alone: meaning it takes on the values observed car by car.

```
. adjust mpg weight, by(foreign) gen(pred err)
-------------------------------------------------------------------------------
Created variables: pred, err
Variable left as is: turn
Covariates set to mean: mpg = 21.297297, weight = 3019.4595
-------------------------------------------------------------------------------

----------+-----------------------
 Car type |         xb        S.E.
----------+-----------------------
 Domestic |   4796.421    564.7897
  Foreign |   9400.686    643.9684
----------+-----------------------
```

Specifying the `gen(pred err)` generated prediction and error variables that we can use subsequently. Lets take a look at a few observations and compare them to the actual price and the predicted price using `predict` without any constraints.

```
. predict pred2 , xb

. list foreign turn price pred2 pred err in 47/58
      foreign      turn     price       pred2        pred         err
 47. Domestic        42      5798     7585.93    4668.595     373.002
 48. Domestic        42      4934    6600.487    4668.595     373.002
 49. Domestic        45      5222    4798.425    3980.978    628.3688
 50. Domestic        40      4723    5901.609    5127.007    314.9057
 51. Domestic        43      4424    6156.588    4439.389    445.2716
 52. Domestic        41      4172    3484.962    4897.801    325.7249
 53.  Foreign        37      9690    8226.297     9036.04    548.7275
 54.  Foreign        36      6295    5196.463    9265.246    589.1915
 55.  Foreign        34      9735    8138.971    9723.657    719.4052
 56.  Foreign        35      6229    6711.028    9494.451    647.5291
 57.  Foreign        32      4589    5893.468    10182.07    890.1414
 58.  Foreign        34      5079     6554.16    9723.657    719.4052
```

You can see that there are differences in the predictions from the `predict` command and those from `adjust`. `predict` uses each observation's individual values for the variables in the regression model. `adjust` substitutes certain values for some or all of the variables in the regression model depending on what was specified in the `adjust` command. The first produces predicted values for the cars. The second produces predicted values for the cars with certain characteristics changed.

Say we wish to look at the predicted cost of domestic and foreign cars by repair record (`rep78`) under the current regression model holding `mpg` and `weight` to their mean and allowing `turn` to vary. `adjust` allows us to do this even though we have not included `rep78` in the regression model.

```
. adjust mpg weight, by(foreign rep78)
-------------------------------------------------------------------------------
Variable left as is: turn
Covariates set to mean: mpg = 21.289856, weight = 3032.0291
-------------------------------------------------------------------------------
Table cells contain: xb and S.E.

----------+-------------------------------------------------
          |                Repair Record 1978
 Car type |        1         2         3         4         5
----------+-------------------------------------------------
 Domestic | 4951.659  4407.295  4790.366  4722.454  6097.688
          | 341.7415  541.8098  567.7991  579.3939   590.864
          |
  Foreign |                     9471.907   9548.31  9420.973
          |                      641.199  673.0413  639.7968
----------+-------------------------------------------------
```

This is an example of a two-way table produced by `adjust`. Up to seven-way tables are possible.

You may have noticed that the means of `mpg` and `weight` in this run of `adjust` are not the same as in the previous runs. This is due to the fact that the `rep78` variable (which was not a part of the regression) has five missing values and `adjust` does casewise deletion when confronted with missing values.

What if we wanted to find the predicted cost of foreign and domestic cars under this regression model while setting `weight` to its mean, letting `mpg` and `turn` vary, and pretending for the moment that all of the cars are domestic? `adjust` handles this since it allows variables to be set to a specific value and also used in the `by` option.

```
. adjust weight foreign=0, by(foreign)
```

```
-------------------------------------------------------------------------------
Variables left as is: mpg, turn
Covariate set to mean: weight = 3019.4595
Covariate set to value: foreign = 0
-------------------------------------------------------------------------------

----------+-----------------------
 Car type |         xb        S.E.
----------+-----------------------
 Domestic |   4797.107    645.4536
  Foreign |   6177.651    803.4382
----------+-----------------------
```

In this table we are obtaining the predicted prices of all cars as if they were domestic. The $6,177 prediction, for instance, is the average predicted price of our sample of foreign cars were they instead domestic and if they had average weight. The foreign-car sample has a different prediction than the domestic-car sample because the cars in the two samples have different `mpg` and `turn` and we left these differences just as we observed them.

We now do the same thing except we treat all cars as if they were foreign by typing

```
. adjust weight foreign=1, by(foreign)
-------------------------------------------------------------------------------
Variables left as is: mpg, turn
Covariate set to mean: weight = 3019.4595
Covariate set to value: foreign = 1
-------------------------------------------------------------------------------

----------+-----------------------
 Car type |         xb        S.E.
----------+-----------------------
 Domestic |   8018.522    823.5443
  Foreign |   9399.066    841.2903
----------+-----------------------
```

Put either way, the tables report the same difference in prices due to `mpg` and `turn` between the domestic and foreign car samples: $9399.066 - 8018.522 = 1380.544$ or $6177.651 - 4797.107 = 1380.544$.

The output from these two `adjust` commands can be compared with the following:

```
. adjust weight, by(foreign)
-------------------------------------------------------------------------------
Variables left as is: mpg, turn
Covariate set to mean: weight = 3019.4595
-------------------------------------------------------------------------------

----------+-----------------------
 Car type |         xb        S.E.
----------+-----------------------
 Domestic |   4797.107    645.4536
  Foreign |   9399.066    841.2903
----------+-----------------------
```

Example 2

What if we decide to include the `rep78` variable in the regression model of Example 1? An easy way to do that is with `xi`.

```
. xi : regress price mpg weight turn I.rep78
I.rep78           Irep78_1-5   (naturally coded; Irep78_1 omitted)
  Source |       SS       df       MS                  Number of obs =      69
---------+------------------------------               F(  7,    61) =    6.34
   Model |  242819042     7  34688434.6                Prob > F      =  0.0000
Residual |  333977917    61  5475047.82                R-squared     =  0.4210
---------+------------------------------               Adj R-squared =  0.3545
   Total |  576796959    68  8482308.22                Root MSE      =  2339.9

------------------------------------------------------------------------------
   price |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
     mpg |  -86.24749   84.98694     -1.015   0.314      -256.1894     83.6944
  weight |    3.39851   .8279604      4.105   0.000       1.742901     5.05412
    turn |  -321.7209   136.6736     -2.354   0.022      -595.0167   -48.42516
Irep78_2 |   1143.126    1860.47      0.614   0.541      -2577.113    4863.365
Irep78_3 |   1074.757   1715.121      0.627   0.533       -2354.84    4504.353
Irep78_4 |   1541.853   1759.521      0.876   0.384      -1976.527    5060.234
Irep78_5 |   2813.323   1849.747      1.521   0.133      -885.4748    6512.121
   _cons |   9030.873   5599.464      1.613   0.112      -2165.946    20227.69
------------------------------------------------------------------------------
```

Now we wish to return to our two-way adjusted table in Example 1 and perform the same kind of adjusted predictions under this new regression model. We will set mpg and weight to their mean values and allow turn to vary and obtain the predictions for domestic and foreign cars by repair record.

```
. adjust mpg weight, by(foreign rep78)
-------------------------------------------------------------------------------
Variables left as is: turn, Irep78_2, Irep78_3, Irep78_4, Irep78_5
Covariates set to mean: mpg = 21.289856, weight = 3032.0291
-------------------------------------------------------------------------------
Table cells contain: xb and S.E.
----------+-----------------------------------------------
          |                  Repair Record 1978
 Car type |        1          2          3          4          5
----------+-----------------------------------------------
 Domestic | 4308.501    4687.54   5156.861   5528.633   8730.428
          | 1660.918   919.0236   737.6397     863.02   891.2327
          |
  Foreign |                       7206.343    7780.68   8873.416
          |                       769.9142   821.4432   930.6635
----------+-----------------------------------------------
```

adjust can take advantage of the original rep78 variable in the by option. You will notice that the output says that the xi created variables are left as is along with the turn variable. This is true, but with the rep78 variable in the by option adjust still produces the results we desire.

If you have used xi in your estimation command, you can freely use the original variables that xi operated on in the by option. The same is not true for setting these variables to specific values. In that case you must use the names produced by xi. As an example lets say that we wish to create the same adjusted prediction table as before but we now want to treat all the data as if it had a repair record of 3. Here is how we would do it.

```
. adjust mpg weight Irep78_2=0 Irep78_3=1 Irep78_4=0 Irep78_5=0, by(foreign rep78)
-------------------------------------------------------------------------------
Variable left as is: turn
Covariates set to mean: mpg = 21.289856, weight = 3032.0291
Covariates set to value: Irep78_2 = 0, Irep78_3 = 1, Irep78_4 = 0, Irep78_5 = 0
-------------------------------------------------------------------------------
Table cells contain: xb and S.E.
----------+-----------------------------------------------
          |                  Repair Record 1978
 Car type |        1          2          3          4          5
----------+-----------------------------------------------
 Domestic | 5383.258    4619.17   5156.861   5061.537   6991.862
          | 488.3862   727.8113   737.6397   753.6007   700.4694
          |
  Foreign |                       7206.343   7313.583   7134.849
          |                       769.9142     819.66   759.8754
----------+-----------------------------------------------
```

If you wanted to do the same thing except set the repair record to 1 (the level dropped by xi), then in the adjust command you set all the xi created variables to zero.

```
. adjust mpg weight Irep78_2=0 Irep78_3=0 Irep78_4=0 Irep78_5=0, by(foreign rep78)
```

(*output omitted*)

And finally for this example, we demonstrate a three-way table produced by adjust.

```
. gen turngrp = autocode(turn,3,30,51)
. label define tgrp 37 "31-37" 44 "38-44" 51 "42-51"
. label val turngrp tgrp
. label var turngrp "Turn Circle (ft.)"
```

(*Continued on next page*)

```
. adjust mpg weight, by(rep78 foreign turngrp)
-------------------------------------------------------------------------------
Variables left as is: turn, Irep78_2, Irep78_3, Irep78_4, Irep78_5
Covariates set to mean: mpg = 21.289856, weight = 3032.0291
-------------------------------------------------------------------------------
Table cells contain: xb and S.E.
----------+----------------------------------------------------------------
Repair    |                  Turn Circle (ft.) and Car type
Record    | ------ 31-37 -----    ------ 38-44 -----    ------ 42-51 -----
1978      | Domestic   Foreign    Domestic   Foreign    Domestic   Foreign
----------+----------------------------------------------------------------
        1 |                       4308.501
          |                       1660.918
          |
        2 |                        5194.25              3843.022
          |                       861.8954              1007.061
          |
        3 | 7474.443  7206.343    5044.604              3372.501
          | 909.8427  769.9142    564.0601              1154.845
          |
        4 | 8424.121  7901.325    5528.633  6815.517    4080.889
          | 1012.321  846.7875    704.0888  580.1096    1162.753
          |
        5 | 8730.428  8873.416
          | 891.2327  930.6635
----------+----------------------------------------------------------------
```

Example 3

To show the use of `adjust` in producing probabilities, we model the `foreign` variable as a function of a few of the other variables.

```
. logit foreign weight mpg
( iteration log omitted )
Logit Estimates                                         Number of obs =     74
                                                        chi2(2)       =  35.72
                                                        Prob > chi2   = 0.0000
Log Likelihood = -27.175156                             Pseudo R2     = 0.3966
------------------------------------------------------------------------------
 foreign |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
  weight |  -.0039067   .0010116     -3.862   0.000      -.0058894    -.001924
     mpg |  -.1685869   .0919174     -1.834   0.067      -.3487418     .011568
   _cons |   13.70837   4.518707      3.034   0.002       4.851864    22.56487
------------------------------------------------------------------------------
```

We can obtain the predicted probability of foreign manufacture by repair record under this logit model setting `mpg` to its mean and allowing `weight` to vary.

```
. adjust mpg, by(rep78) pr
-------------------------------------------------------------------------------
Variable left as is: weight
Covariate set to mean: mpg = 21.289856
-------------------------------------------------------------------------------
----------+-----------
Repair    |
Record    |
1978      |          P
----------+-----------
        1 |   .1989613
        2 |   .1173622
        3 |   .2050817
        4 |   .4636117
        5 |   .6904407
----------+-----------
```

If the `pr` option were not specified, we would have obtained the `xb` results.

```
. adjust mpg, by(rep78)
-------------------------------------------------------------------------------
Variable left as is: weight
Covariate set to mean: mpg = 21.289856
-------------------------------------------------------------------------------
```

```
----------+-----------------------
Repair    |
Record    |
1978      |         xb        S.E.
----------+-----------------------
        1 |  -1.991596    .6367775
        2 |  -2.982921    .8305857
        3 |  -2.769029    1.003742
        4 |  -1.093055    .9754658
        5 |   1.044976    .6983151
----------+-----------------------
```

It is important to remember that the default for `adjust` is `xb` and that `pr` is available after only a handful of estimation commands (as provided by `predict`).

References

Garrett, J. M. 1998. sg33.1: Enhancements for calculation of adjusted means and adjusted proportions. *Stata Technical Bulletin* 43: 16–24.

ssa12 Predicted survival curves for the Cox proportional hazards model

Joanne M. Garrett, University of North Carolina at Chapel Hill, FAX 919-966-2274, garrettj@med.unc.edu

After fitting a Cox proportional hazards model, we often want to graph the survival probabilities by different categories of a nominal independent variable, adjusted for other variables in the model. These predictions are based on a single baseline hazard function, and visually represent the hazard ratio estimated by the Cox model. In Stata, `sts graph` gives us graphs based on a stratified Cox model (with the `strata` option), using the Kaplan–Meier estimates derived from separate baseline hazards for these categories (see [R] **sts graph**). If a variable violates the proportional hazards assumption, the stratified model and the Kaplan–Meier graph would be the appropriate result to report. However, if we want to report hazard ratios and not stratify, we may prefer a graph that displays the predicted results. A program which does this is called `stcoxplt`.

Syntax

`stcoxplt` [`if` *exp*] , `xvar(`*xvar*`)` [`adjust(`*varlist*`)` `model` `nolog` `noshow` *graph_options*]

Options

`xvar(`*xvar*`)` is a nominal independent variable; separate predicted survival curves are plotted for categories of this variable. Dummy variables are created with the lowest category defaulting to the reference group. These variables are named `x2`, `x3`, etc.

`adjust(`*varlist*`)` adjusts the estimates for covariates; any variables listed are centered by default before estimation based on the observations used in the model, or they may be set to a specified value (e.g., `age=40 gender=1`).

`model` displays the output for the Cox model; by default, `model` results are not displayed.

`nolog` suppresses the iteration log.

`noshow` prevents `stcoxplt` from showing the key `st` variables.

graph_options are most of the options available in `graph`; the *y*-axis is labeled by the program; If `symbol()` and `connect()` are not specified, default symbols are used and points are not connected.

A second program included in this insert is `stkap.ado`, which graphs the adjusted Kaplan–Meier estimates. The only difference between this program and `sts graph` is the adjustment variables are centered automatically, or set to user-specified values.

Syntax

`stkap` [`if` *exp*] [, `by(`*varlist*`)` `strata(`*varlist*`)` `adjust(`*varlist*`)` *graph_options*]

Options

`by(`*varlist*`)` graphs a separate curve for each category of *varlist*; produces separate survivor functions by making separate calculations for each group identified by equal values of the `by()` variables.

`strata(`*varlist*`)` graphs a separate curve for each category of *varlist*; stratified estimates (equal coefficients across strata but baseline hazard unique to each stratum) are then estimated. May be used with `adjust()` in place of `by()`.

`adjust(`*varlist*`)` adjusts the estimates for covariates; any variables listed are centered by default before estimation based on the observations used in the model, or they may be set to a specified value (e.g., `age=40`, `gender=1`).

graph_options are most of the options available in `graph`.

Examples

All the examples come from a leukemia remission study. The data consist of 42 patients who are followed over time to see how long (`weeks`) before they go out of remission (`relapse`: 1 = yes, 0 = no). Half the patients receive a new experimental drug treatment and the other half receive the standard drug (`trtment1`: 1 = drug A, 0 = standard). White blood cell count, strongly related to the presence of leukemia, is defined two ways: as the natural log of the white blood cell count (`lnw bcc`) and divided into three categories (`wbc3cat`: 1 = normal, 2 = moderate, 3 = high).

```
. use leukemia

. describe

Contains data from leukemia.dta
  obs:           42                     Leukemia Remission Study
 vars:            6                     19 Feb 1998 10:31
-------------------------------------------------------------------
   1. weeks      byte   %8.0g           Weeks in Remission
   2. relapse    byte   %8.0g   yesno     Relapse
   3. trtment1   byte   %8.0g   trt1lbl   Treatment I
   4. trtment2   byte   %8.0g   trt2lbl   Treatment II
   5. lnwbcc     float  %9.0g           Ln White Blood Cell Count
   6. wbc3cat    byte   %9.0g   wbclbl    White Blood Cell Count
-------------------------------------------------------------------

. stset weeks relapse   (as with all st analyses, data must be stset first)

   data set name:  leukemia.dta
              id:  --              (meaning each record a unique subject)
      entry time:  --              (meaning all entered at time 0)
       exit time:  weeks
   failure/censor:  relapse
```

Example 1

We graph the predicted survival curves for `weeks` to `relapse` by treatment category (Drug A versus standard), adjusted for `lnwbcc`. We use the default symbols, connected with steps, and print the results of the Cox model. Dummy variables a re created for `trtment1`. Since `trtment1` has two categories, only one dummy variable is needed for the model, and is named `x2`.

```
. stcoxplt, xvar(trtment1) adjust(lnwbcc) xlabel c(JJ) model nolog

    failure time:  weeks
  failure/censor:  relapse

Cox regression -- entry time 0

No. of subjects =   42                    Log likelihood =   -72.27926
No. of failures =   30                    chi2(2)        =       43.41
Time at risk    =  541                    Prob > chi2    =      0.0000

-----------------------------------------------------------------------
   weeks |
 relapse | Haz. Ratio   Std. Err.    z     P>|z|   [95% Conf. Interval]
---------+-------------------------------------------------------------
      x2 |   .2741535   .1157213  -3.066   0.002   .1198666    .6270315
  lnwbcc |   4.974592   1.638274   4.872   0.000   2.608751     9.48598
-----------------------------------------------------------------------
```

The dummy variable `x2` for `trtment1` gives us the hazard ratio comparing patients on Drug A (`trtment1` = 1) versus patients on the standard drug (`trtment1` = 0), adjusted for `lnwbcc`. We see, both from the graph in Figure 1 and the model, that patients on Drug A relapse at a much slower rate than patients on the standard drug treatment.

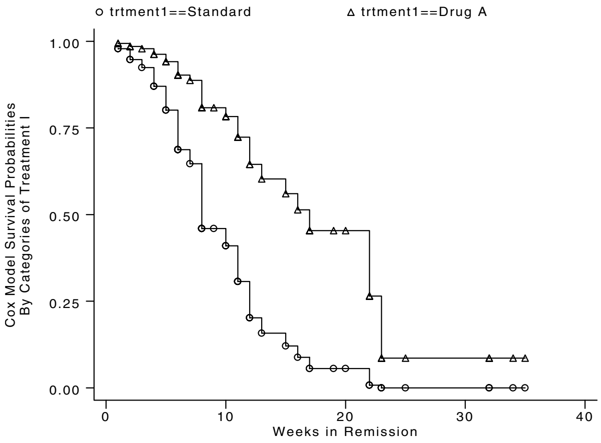

Figure 1. Predicted survival curves for `weeks` to `relapse` by treatment category (Drug A versus standard), adjusted for `lnwbcc`

The Kaplan–Meier version of this plot is displayed in Figure 2.

```
. stkap, strata(trtment1) adjust(lnwbcc)
```

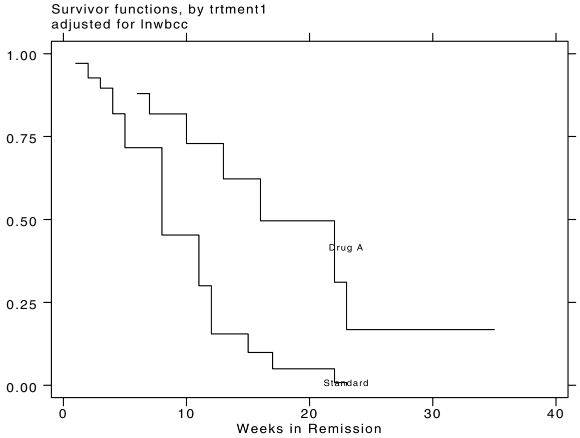

Figure 2. Kaplan–Meier version of the plot in Figure 1.

The two graphs look very similar, since `trtment1` does not violate the proportional hazards assumption. There are subtle differences, however. The Cox model predicts survival probabilities for all time points for each group, whether there are failures at those time points or not. For instance, notice in the Kaplan–Meier graph in Figure 2 that there are no failures for Drug A before 6 weeks or estimates for the standard drug after 23 weeks. This happens because no patients in the Drug A group relapse until 6 weeks, and all the patients in the standard drug group have relapsed by 23 weeks. However, in Figure 1 we see predicted survival probabilities for these time points, even though observed failures have not occurred.

Example 2

We graph the predicted survival curves for `weeks` to `relapse` by white blood cell count in three categories (`wbc3cat`: 1 = normal, 2 = moderate, 3 = high), adjusted for `trtment1`. We use the default symbols, connected with steps, and print the results of the Cox model. Two dummy variables are used for `wbc3cat`. Normal white blood cell count defaults to the reference category, `x2` compares relapse rates for patients with moderately elevated white blood cell counts (`wbc3 cat` = 2) to normal patients, and `x3` compares relapse rates for patients with high white blood cell counts (`wbc3cat` = 3) to normal patients.

(*Continued on next page*)

```
. stcoxplt, xvar(wbc3cat) adj(trtment1) xlabel c(JJJ) model nolog

    failure time:  weeks
  failure/censor:  relapse

Cox regression -- entry time 0

No. of subjects =   42                    Log likelihood =  -77.476905
No. of failures =   30                    chi2(3)        =       33.02
Time at risk    =  541                    Prob > chi2    =      0.0000

------------------------------------------------------------------------
   weeks |
 relapse | Haz. Ratio   Std. Err.    z     P>|z|   [95% Conf. Interval]
---------+--------------------------------------------------------------
      x2 |   3.637825   2.201306   2.134   0.033   1.111134    11.91015
      x3 |   10.92214   7.088783   3.684   0.000    3.06093    38.97284
trtment1 |   .2834551   .1229874  -2.906   0.004   .1211042    .6634517
------------------------------------------------------------------------
```

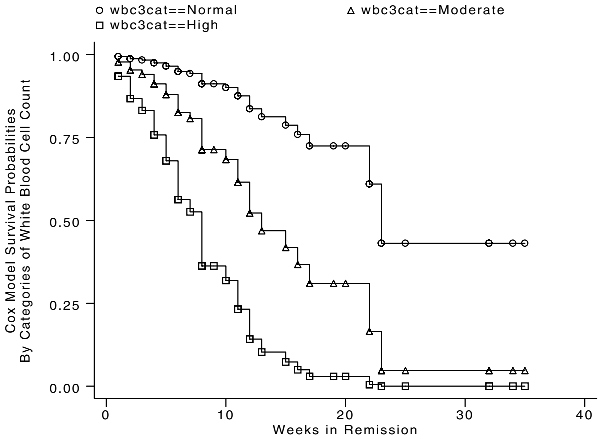

Figure 3. Predicted survival curves for `weeks` to `relapse` by white cell counts adjusted for `trtment1`.

As we can see from Figure 3 and the hazard ratios in the Cox regression table, higher levels of white blood cell counts dramatically increase the rate of relapse.

Example 3

There is another variable in this dataset which measures a different drug (`trtment2`: 1 = drug B, 0 = standard). This example models `trtment2`, adjusted for `lnwbcc`, and graphs the predicted survival curves for `weeks` to `relapse` by treatment category (Drug B versus standard). The variable `x2` compares Drug B to the standard.

```
. stcoxplt, xvar(trtment2) adj(lnwbcc) xlabel c(JJ) model nolog

    failure time:  weeks
  failure/censor:  relapse

Cox regression -- entry time 0

No. of subjects =    42                    Log likelihood =  -77.306546
No. of failures =    30                    chi2(2)        =       33.36
Time at risk    =   541                    Prob > chi2    =      0.0000

------------------------------------------------------------------------
   weeks |
 relapse | Haz. Ratio   Std. Err.    z     P>|z|   [95% Conf. Interval]
---------+--------------------------------------------------------------
      x2 |   .7911487   .3228452  -0.574   0.566   .3555552    1.760391
  lnwbcc |   4.997071   1.500524   5.358   0.000   2.774062    9.001501
------------------------------------------------------------------------
```

(*Continued on next page*)

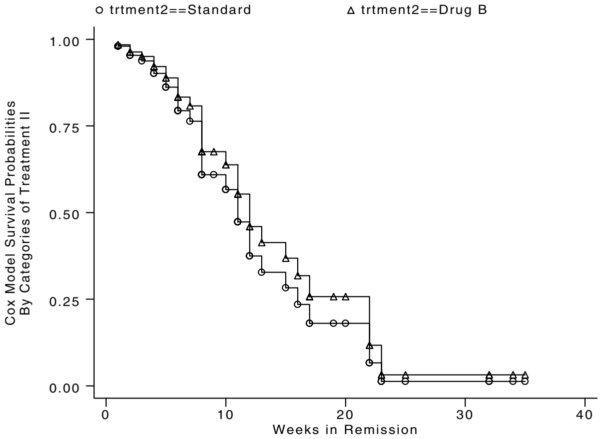

Figure 4. `weeks` to `relapse` by treatment category.

Although the hazard ratio is less than one (HR = 0.79), implying patients are going out of remission at a slower rate when on Drug B, the estimate is not significant ($p = 0.566$). This is confirmed in Figure 4 where the survival curves are very similar for Drug B versus the standard. However, when we look at the Kaplan–Meier graph, we reach a different conclusion.

```
. stkap, strata(trtment2) adj(lnwbcc)
```

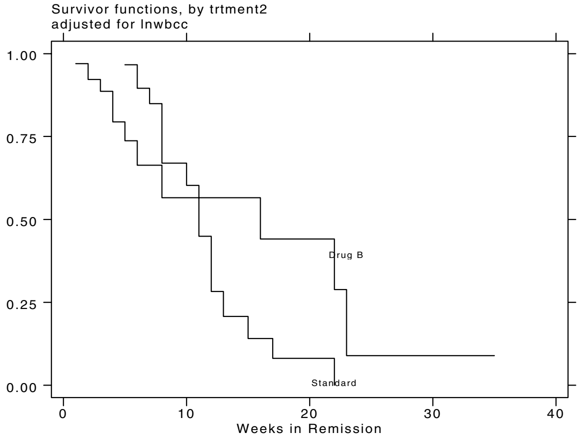

Figure 5. Kaplan–Meier version of Figure 4.

It appears in this graph that patients on Drug B go out of remission more quickly in the earlier weeks of the study (before week 10), but for patients still in remission at 10 weeks, Drug B may work better than the standard drug. The survival curves cross. We don't see this pattern in Figure 4, where every point estimate for the standard treatment is below the point estimate for Drug B—a constraint imposed by the model. This is an example where reporting the hazard ratio and graph from the Cox model would be misleading.

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
Nazerath-Ellit 17100
Israel
Phone: +972-4-9541153
Fax: +972-6-6554254
Email: sasconsl@actcom.co.il
Countries served: Israel

Company: Dittrich & Partner Consulting
Address: Prinzenstrasse 2
D-42697 Solingen
Germany
Phone: +49-212-3390 200
Fax: +49-212-3390 295
Email: evhall@dpc.de
URL: http://www.dpc.de
Countries served: Germany, Austria, Italy

*For the most up-to-date list
of Stata distributors, visit
http://www.stata.com*

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 1 42 46 00 42
Fax: +33 1 42 46 00 33
Email: info@ritme.com
URL: http://www.ritme.com
Countries served: France, Belgium,
Luxembourg, Switzerland

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

(*List continued on next page*)

International Stata Distributors

(*Continued from previous page*)

Company: Survey Design & Analysis
Services P/L
Address: 249 Eramosa Road West
Moorooduc VIC 3933
Australia
Phone: +61-3-59788329
Fax: +61-3-59788623
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: +44 181 4620495
Fax: +44 181 4620493
Email: info@timberlake.co.uk
URL: http://www.timberlake.co.uk
Countries served: United Kingdom, Eire

Company: Timberlake Consulting S.L.
Address: Calle Montecarmelo n°36 Bajo
41011 Seville
Spain
Phone: +34.5.428.40.94
Fax: +34.5.428.40.94
Email: timberlake@zoom.es
Countries served: Spain

Company: Timberlake Consultores, Lda.
Address: Praceta do Comércio, 13 - 9° Dto.
Quinta Grande
2720 Alfragide
Portugal
Phone: +351 (0)1 471 9337
Fax: +351 (0)1 471 9337
Telemóvel: +351 (0)931 6272 55
Email: timberlake.co@mail.telepac.pt
Countries served: Portugal

Company: Unidost A.S.
Address: Rihtim Cad. Polat Han D:38
Kadikoy
81320 ISTANBUL
Turkey
Phone: +90-216-4141958
Fax: +90-216-3368923
Email: info@unidost.com
URL: http://abone.turk.net/unidost
Countries served: Turkey