STATA TECHNICAL BULLETIN

July 1997
STB-38

A publication to promote communication among Stata users

Contents of this issue

dm48 An enhancement of reshape

Jeroen Weesie, Utrecht University, Netherlands, weesie@weesie.fsw.ruu.nl

`reshape` is a very useful Stata command to convert cross-sectional time-series information and other forms of multilevel data between a wide storage format (different measurements that belong to a common unit are stored in different variables for the unit) and a long storage format (each measurement on each unit is stored as a separate observation). The current implementation of `reshape` suffers from some limitations. First, the number of "constant" (level 1) variables in `reshape` is restricted to 10. Second, `reshape` assumes that the names of the variables that contain the related measurements in wide format follow a mask "*name*|*nr*", in which *nr* takes integer values only.

The program `reshape2` described in this insert seeks to eliminate these limitations, while remaining fully backward compatible with `reshape`. In fact, `reshape2` is a fairly extensive rewrite of the code of `reshape`. Thus, the keyword-based syntax of `reshape2` is maintained, even though I would have preferred a syntax that is consistent with the standard Stata command syntax in which information is transferred via arguments (options). To increase the number of variables constant in long format, the constant variables are split into unit-1 identification variables specified via the new keyword `id`, and other "constant" variables. While each unit-1 observation should have unique values on the identification variable(s), this is of course not required for the constant variables. To allow for more general masks for the names of variables of related measurements, the user should specify masks in which an `@` should be replaced by the group variable. Again for backward compatibility, if the mask does not contain a `@`, one is silently appended. The other modifications required only changes to the internal "logic" of `reshape`.

Syntax

The syntax of `reshape2` is

```
reshape2 clear

reshape2 id varname [ varname ... ]

reshape2 cons varname [ varname ... ]

reshape2 groups groupvar #[-#] [#[-#] ... ] [, long(string) string ]

reshape2 vars maskname [ maskname ... ]

reshape2 { wide | long }

reshape2 query
```

Description

`reshape2` converts data from wide to long and vice-versa.

`reshape2` assumes that the names of variables for which there are related observations fit masks (see keyword `vars` below) in which the placeholder `@` is replaced by a set of values (see `groups` below) in wide format and by a single character in long format (see the option `long` for the `groups` keyword).

`reshape2 clear` clears the current definition elements.

`reshape2 id` specifies the case-identification variable(s) (e.g., the respondent number). In wide format, the `id` variable(s) should strictly vary between observations. For compatibility with `reshape`, if `reshape2 id` is not specified, the `cons` variables are used as identification variables. The separation of `cons` variables into `id` variables and "other" `cons` variables allows `reshape2` to process data manipulation with many `cons` variables, whereas `reshape` was limited to 10 `cons` variables.

`reshape2 cons` identifies the variable(s) that are "relatively" constant; that is, that do not change across related observations.

`reshape2 groups` names a single variable that will record the grouping variable along with the values it will assume. The grouping variable is the variable that will be created when converting from wide to long and the values are the values it will assume, separated by blanks. If the option `string` is specified, the values are interpreted as strings. Otherwise, the values should be positive integers, and the specification of the group values may include numeric ranges.

`reshape2 vars` identifies a list of masks, separated by white space, for each of the variable(s) for which there are related observations. A mask should contain at most one place holder `@`. If a mask does not contain a `@`, a `@` is silently appended to the mask. Actually, a keyword `mask` would better describe the function of this subcommand. For compatibility with `reshape`, we use the name `vars`.

`reshape2 long` converts data to long format.

`reshape2 wide` converts to wide.

`reshape2 query` displays the current definitions.

Example 1

Our first example is the same as the one for `reshape` in the Stata manual:

```
id  sex   inc80   inc81   inc82        id   year   sex    inc
------------------------------         ---------------------
 1    0    5000    5500    6000         1     80     0   5000
 2    1    2000    2200    3300         1     81     0   5500
 3    0    3000    2000    1000         1     82     0   6000
                                        2     80     1   2000
                                        2     81     1   2200
                                        2     82     1   3300
                                        3     80     0   3000
                                        3     81     0   3300
                                        3     82     0   1000
. reshape2 groups year 80-82
. reshape2 vars inc
. reshape2 cons id sex
. reshape2 long                    (goes from left-form to right)
. reshape2 wide                    (goes from right-form to left)
```

Example 2: Three-level data

We now illustrate how `reshape2` can be used for the manipulation of data that contains more than two levels. While the current implementation of `reshape2` does not support the description of 3-level data, two simple `reshape2` steps will get the job done. Suppose we have a data set on households, each of which has a male and female spouse, a number of children, and for each household we have the variables `hnr` (the number of the household), `hcity` (the city where the household is located), `hnkids` (the number of children in the household), `medu`, `fedu` (the education level of the male and female spouse, respectively), `minc90`, `minc91` (the 1990 and 1991 income of the male spouse), and `finc90`, `finc91` (the 1990 and 1991 income of the female spouse). We will think of this arrangement of the data as in wide-wide format. Here is a simple data set consisting of three observations:

```
hnr    hcity     hnkids    medu  minc90     minc91     fedu  finc90     finc91
  1       NY          3       B   30000      32000        H   23000      23700
  2     Phil          1       B   31000      33100        B   34200      35200
  3       SF          2       M   43000      45100        B   35000      37250
```

Now suppose we want to reshape the data set so that there is an observation for each of the spouses in each household ("long-wide" format):

```
. reshape2 id hnr
. reshape2 cons hcity hnkids
. reshape2 groups sex m f, string
. reshape2 vars @inc90 @inc91 @edu
. reshape2 long
. sort hnr
. list, nodisplay noobs
      hnr      hcity     hnkids      inc90      inc91        edu        sex
        1         NY          3      30000      32000          B          m
        1         NY          3      23000      23700          H          f
        2       Phil          1      34200      35200          B          f
        2       Phil          1      31000      33100          B          m
        3         SF          2      35000      37250          B          f
        3         SF          2      43000      45100          M          m
```

Finally, we can reshape the data again so that there is an observation for each income for each spouse ("long-long" format):

```
. reshape2 id hnr sex
. reshape2 cons hcity hnkids edu
. reshape2 groups year 90-91
. reshape2 vars inc@
. reshape2 long
. sort hnr sex year
. list, nodisplay noobs
        hnr      hcity     hnkids        edu        sex        inc       year
          1         NY          3          H          f      23000         90
          1         NY          3          H          f      23700         91
          1         NY          3          B          m      30000         90
          1         NY          3          B          m      32000         91
          2       Phil          1          B          f      34200         90
          2       Phil          1          B          f      35200         91
          2       Phil          1          B          m      31000         90
          2       Phil          1          B          m      33100         91
          3         SF          2          B          f      35000         90
          3         SF          2          B          f      37250         91
          3         SF          2          M          m      43000         90
          3         SF          2          M          m      45100         91
```

sbe15 Age-specific reference intervals for normally distributed data

Eileen Wright, Royal Postgraduate Medical School, UK, ewright@rpms.ac.uk
Patrick Royston, Royal Postgraduate Medical School, UK, proyston@rpms.ac.uk

Reference intervals (RIs) are routinely used in medicine to determine whether values are "normal" or "abnormal." Values lying outside the limits of the interval are classed as "abnormal." The measurements of interest may be known to be dependent on age; the limits of an age-specific RI are then defined by curves. As an example, Figure 1 shows a 95% RI (i.e. the estimated 2.5th and 97.5th centile curves) and median for fetal biparietal diameter (a measurement of head size) by gestational age. The biparietal diameter is a measurement of head size, in this case calculated between the proximal edges of the fetal skull at the deep borders of the ultrasound beam. The data-set is available on the STB-38 disk in **bpd.dta** and is described in more detail by Chitty et al. (1994).

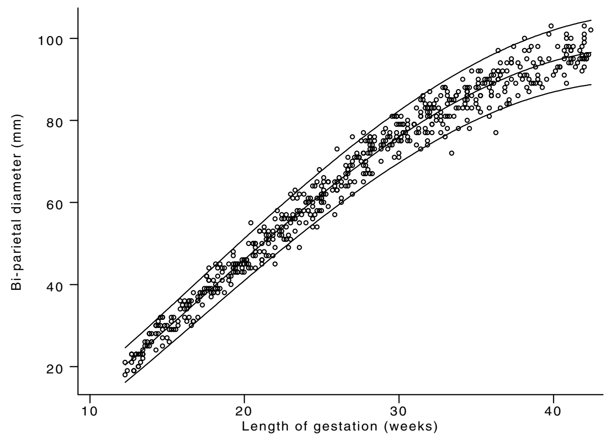

Figure 1. 95% RI and median for fetal biparietal diameter against gestational age.

Many measurements, particularly those of fetal size observed in ultrasound scans, are adequately modeled by a normal distribution (Royston and Wright 1997a) conditional on age. Figure 1 was obtained with the software presented here (**xrigls**), which finds suitable fractional polynomials (FPs) (see [R] **fracpoly** and Royston and Altman 1994) for the age-specific mean and standard deviation (SD) curves. It uses an iterative procedure (generalized least squares or GLS). The analysis is as follows:

```
. xrigls bpd gawks, fp(m:df 4,s:df 2) centile(2.5 97.5) detail

          --- FP Powers ---
Cycle     Mean       SD          Deviance     Change  Residual SS
-----------------------------------------------------------------
0         2 2                    3021.644      0.000   5708.761
1         2 2        1           2988.312    -33.332    5094.04
2         2 2        1           2988.320      0.008   5094.916

Final deviance =  2988.320 (592 observations).
Power(s) for mean curve = 2 2. Power(s) for SD curve = 1.
```

```
Regression for mean curve
-------------------------
(sum of wgt is  6.8916e+001)
  Source |       SS       df       MS                  Number of obs =     592
---------+------------------------------               F(  2,   589) =17178.91
   Model |  297199.013     2  148599.507               Prob > F      =  0.0000
Residual |  5094.91599   589  8.65011204               R-squared     =  0.9831
---------+------------------------------               Adj R-squared =  0.9831
   Total |  302293.929   591  511.49565                Root MSE      =  2.9411

------------------------------------------------------------------------------
     bpd |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
    Xm_1 |    20.2314   .3000757     67.421   0.000       19.64205    20.82075
    Xm_2 |  -10.01732   .1975909    -50.697   0.000      -10.40539   -9.629256
   _cons |  -7.035313   .6528228    -10.777   0.000      -8.317457   -5.753169
------------------------------------------------------------------------------
   4. Xm_1        float  %9.0g                  x^ 2: x = gawks/10
   5. Xm_2        float  %9.0g                  x^ 2*ln(x): x = gawks/10
Regression for SD curve
-----------------------
  Source |       SS       df       MS                  Number of obs =     592
---------+------------------------------               F(  1,   590) =   27.36
   Model |  151.895774     1  151.895774               Prob > F      =  0.0000
Residual |  3276.12205   590  5.55274924               R-squared     =  0.0443
---------+------------------------------               Adj R-squared =  0.0427
   Total |  3428.01782   591  5.80036857               Root MSE      =  2.3564

------------------------------------------------------------------------------
Abs. res |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
    Xs_1 |   .0614736   .0117536      5.230   0.000       .0383896    .0845575
   _cons |   1.389076   .3336352      4.163   0.000       .7338184    2.044333
------------------------------------------------------------------------------
   6. gawks        float  %9.0g                  gawks
```

`xrigls` selects the best fitting powers and the most appropriate degree of FP at each cycle of the GLS procedure. The maximum degrees of freedom for the mean (`m`) and SD (`s`) are specified in the `fp` option. The significance level used to determine the most appropriate FP for each parameter is 0.05 by default but may be specified using the `alpha` option. As well as plotting the centiles and median superimposed on the raw data, the software creates variables which contain the estimated mean (`M_gls`), SD (`S_gls`) and standard deviation or Z scores (`Z_gls`). If the model is appropriate, the Z scores are approximately normally distributed with mean 0 and SD 1. Variables containing the 3rd and 97th (`C3_gls` and `C97_gls`) centiles are also created by default. Different centiles may be chosen using the `centile` option (or using `centcalc`, see Wright and Royston 1996). The `detail` option displays the regression output for the final estimated mean and SD curves and the details of the FP transformations applied, allowing one to obtain the formula for the curves. For example, the above mean curve is

$$\texttt{mean} = 20.23 - 10.02 \times (\texttt{gawks}/10)^2 - 7.035 \times \log(\texttt{gawks}/10) \times (\texttt{gawks}/10)^2$$

The GLS algorithm alternates between estimating the mean and the standard deviation curves. Consider the measurement of interest Y and corresponding values of age T. In the preliminary cycle (0), the mean is obtained from a least squares regression of Y on T and the SD from a regression of the absolute residuals (see Altman 1993 and Wright and Royston 1996) on T. In subsequent cycles the regression is weighted using the inverse square of the estimated SD curve from the previous cycle. Carroll and Ruppert (1988) point out that it is unnecessary to iterate to convergence; about two cycles are sufficient. In `xrigls` the best-fitting fractional polynomial is found by least squares at each step. Different best powers of T may be selected at each cycle of the procedure, but in practice the powers for the SD curve hardly vary from cycle to cycle and those for the mean curve are stable after the first weighted fit (cycle 1).

The need to obtain a suitable model for the mean curve is perhaps obvious. However, sometimes the need to model the SD curve is overlooked; a constant is assumed and estimated from the residuals of the mean fit. This forces the estimated centile curves to be parallel. However, failure to model heteroscedasticity (age-varying SD) will result in inaccurate estimates of the centile curves, see Figure 2, which is produced by the following code:

```
. xrigls bpd gawks, fp(m:2 2,s:df 0) cent(2.5 97.5) nograph
. for M S Z C2.5 C97.5, any : rename @_gl @_co
. graph bpd C2.5_co M_co C97.5_co gawks, c(.lll) s(oiii) sort xlabel
> ylabel(20,40,60,80,100) gap(5)
```

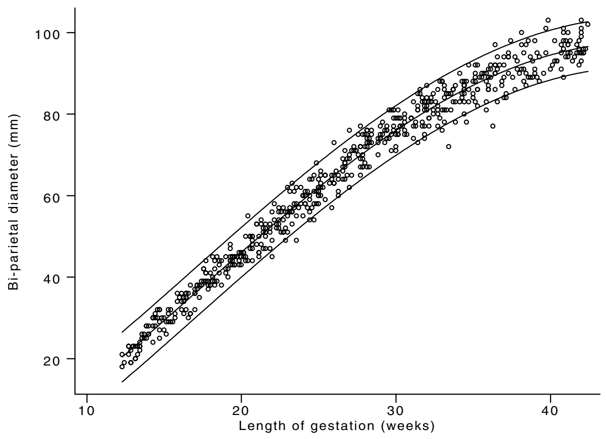

Figure 2. 95% RI and median where SD is an estimated constant.

Here the same FP powers $(2,2)$ have been used for the mean curve and a constant estimated for the SD. The gap between the upper and lower centiles appears to be too wide at low ages. This is more clearly illustrated in a plot of the Z scores (see Figure 3) where the width of the spread of values should be approximately the same across gestational age, but is narrower at low ages.

```
. graph Z_co gawks, xlabel ylabel gap(5)
```

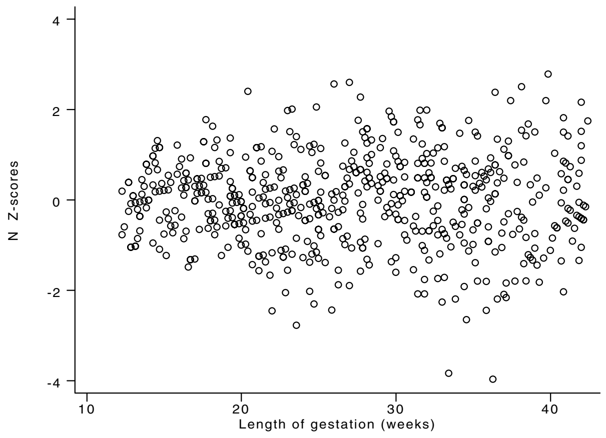

Figure 3. Z scores plotted against gestational age.

In cases where the SD increases markedly with the mean, the coefficient of variation (standard deviation divided by the mean) may be much closer to a constant than the SD itself. Applying the **xrigls** command with the **cv** option and default selection of the FP parts of the model produces the following output:

```
. xrigls bpd gawks, cv centile(2.5 97.5)
              --- FP Powers ---
Cycle      Mean        CV          Deviance     Change  Residual SS
-------------------------------------------------------------------
0          2 2                     3021.644      0.000   5708.761
1          2 2        -2           2985.018    -36.625   4969.406
2          2 2        -2           2984.911     -0.107   4969.078
Final deviance =  2984.911 (592 observations).
Power(s) for mean curve = 2 2. Power(s) for CV curve = -2.
```

Note that the same powers are chosen for the mean curve as before but the inverse square of gestational age is chosen to model the CV. The deviance of this model (2984.91) is slightly lower than that when the SD is modeled (2988.32). Multiplying the CV by its respective mean curve gives an estimate of the SD for this model. The two SD curves are plotted in Figure 4. The new model has a lower SD at low and high gestational ages.

```
. xrigls bpd gawks, fp(m:2 2,s:-2) cv cent(2.5 97.5) nograph
. for M S Z C2.5 C97.5, any : rename @_gl @_cv
. gen SD_cv=S_cv*M_cv
. graph SD_cv S_sd gawks, c(ll) s(ii) so ylabel l1(Standard deviation)
> xlabel gap(5)
```

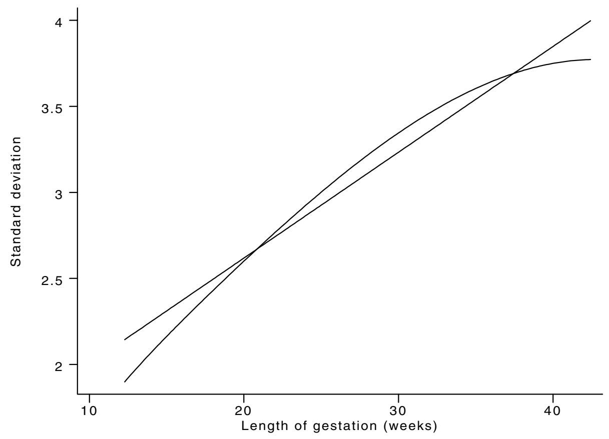

Figure 4. SD estimated directly (full line) and SD estimated from CV (broken line).

However, the estimated 95% RIs for the two sets of results are almost identical (see Figure 5) and the original model, which is slightly simpler, might be preferred.

```
. graph C2.5_cv M_cv C97.5_cv C2.5_sd M_sd C97.5_sd gawks, c(llllll)
> s(iiiiii) sort xlabel ylabel(20,40,60,80,100) l1("Bi-parietal diameter (mm)") gap(5)
```

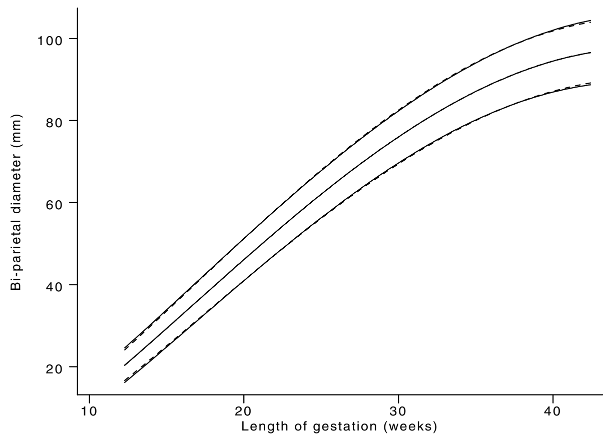

Figure 5. 95% RIs and medians for SD (full lines) and CV (broken lines) models.

The Z scores from the model for the biparietal diameter data with powers $(2,2)$ for the mean and linear SD curve have a p-value of 0.26 from a test of normality (the Shapiro–Wilk W test, `swilk`). When approximate normality is not found, a more complex model may be required. Using exponential transformations and Stata's maximum likelihood `ml` routines, `xriml` (Wright and Royston 1996) fits models which account for non-normal skewness and/or kurtosis in the data.

To gain an impression of the precision of the estimated centile curves, their standard errors may be calculated by the `se` option. Confidence bands of $\pm 2 \times$ standard error are a useful way of illustrating this information. Since the sample size for the fetal biparietal diameter data-set is fairly large, the precision of the estimated centiles is quite high. This is shown in Figure 6 where the confidence bands for the RI and median (see Figure 1) are given for gestational ages greater than 27 weeks.

```
. xrigls bpd gawks, fp(m:2 2,s:1) cent(2.5 50 97.5) nograph se

. for C2.5 C50 C97.5, any : gen l@=@_g-2*se@
. for C2.5 C50 C97.5, any : gen u@=@_g+2*se@

. graph C2.5_gls lC2.5 uC2.5 C50_gls lC50 uC50 C97.5_gl  lC97.5 uC97.5 gawks
> if gawks>27, c(lllllllll) s(iiiiiiiii) xlabel ylabel
> l1("Bi-parietal diameter (mm)") gap(5)
```

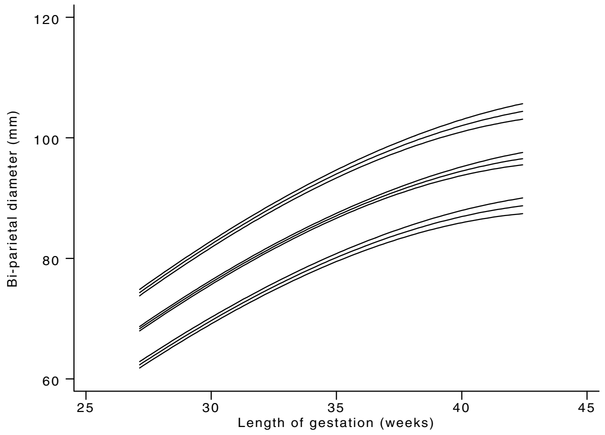

Figure 6. Confidence limits for centiles of biparietal diameter when gestational age is greater than 27 weeks.

Technical note

The variables `M_gls` and `S_gls` created by `xrigls` are the estimated mean and SD curves. When the data are approximately normally distributed, use of `xriml` with the option `dist(n)` gives very similar results to those obtained from `xrigls`. The variables `M_ml` and `S_ml`, created by `xriml`, are also the estimated mean and SD curves. However, when the exponential normal or modulus-exponential normal distributions are selected in `xriml` using the options `dist(en)` or `dist(men)` respectively, `M_ml` and `S_ml` are then the estimated median and scale parameter curves (Wright and Royston 1997b).

Syntax of xrigls

`xrigls` *yvar xvar* [`if` *exp*] [`in` *range*] [`,` *major_options minor_options*]

The *major_options* (most used options) are

`alpha(#)`, `centile(# [# [# ...]])` `cv` `detail` `fp(m:`*term*`,s:`*term*`)`

and *term* is of the form *powers* # | *df* #

The *minor_options* (less used options), in alphabetic order, are

`covars(m:`*mcovars*`,s:`*scovars*`)` `cycles(#)` `nograph` `noleave` `noselect` `notidy` `powers(`*powlist*`)`

`ropts(m:`*mopts*`,s:`*sopts*`)` `saving(`*filename*`[, replace])` `se`

Major options

`alpha(#)` specifies the significance level for testing between degrees of FP for the mean and SD curves. Default is 0.05.

`centile(#` [# [# . . .]]`)` defines the centiles of *yvar* | *xvar* required. Default is 3 and 97 (i.e. a 94% reference interval).

`cv` specifies the `s` curve to be modeled as a coefficient of variation.

`detail` displays the final regression models for the mean and SD curves.

`fp(m:`*term*`,s:`*term*`)` specifies fractional polynomial models in *xvar* for the mean and SD curves. *term* is of the form [*powers*] # [# . . .] | *df* #. The phrase *powers* is optional. The powers should be separated by spaces, for example `fp(m:powers 0 1,s:powers 2)`. If *powers* or *df* are not given for any curve, the default is 4 *df* for the mean and 2 *df* for the SD. # specifies that the degrees of freedom for the best-fitting FP model are to be at most # for the curve in question. The best-fitting powers are then determined from the data.

Minor options

`covars(m:`*mcovars*`,s:`*scovars*`)` includes *mcovars* (*scovars*) variables as predictors in the regression model for the mean (SD) curves.

`cycles(#)` determines the number of fitting cycles. The default value of # is 2: an initial (unweighted) fit for the mean is followed by an unweighted fit of the absolute residuals; weights are calculated, and one weighted fit for the mean, one weighted fit for the absolute residuals, and a final weighted fit for the mean are carried out.

`nograph` suppresses a plot of *yvar* against *xvar* with fitted values and reference limits superimposed. The default is to have the graph.

`noleave` prevents the creation of new variables. The default (`leave`) causes new variables, appropriately labeled, containing the estimated mean, SD and Z scores for *yvar*, also the centiles specified in `centile`, to be created.

`noselect` specifies that the degree of FP will be that specified in the `fp` option. The default is to select a lower order FP if the likelihood ratio test has p-value $<$ `alpha`.

`notidy` preserves the variables created in the routine representing the fractional polynomial powers of the *xvar* used in the analysis.

`ropts(m:`*mopts*`,s:`*sopts*`)` determines the regression options for the mean and SD regression models. For example, `ropt(m:nocons)` suppresses the constant for the mean curve.

`saving(`*filename* `[, replace])` saves the graph to a file (see `nograph`).

`se` calculates the standard errors of the estimated centile curves.

Saved results

`xrigls` saves in `S_#` macros:

`S_1`	deviance of final model
`S_2`	powers in final FP model for mean curve
`S_3`	powers in final FP model for SD curve

Acknowledgments

The project received financial support from grant number 039911/Z/93/Z from The Wellcome Trust. The authors thank Rebecca Turner for her helpful comments on an earlier version of this insert.

References

Altman, D. G. 1993. Construction of age-related reference centiles using absolute residuals. *Statistics in Medicine* 12: 917–924.

Carroll, R. J. and D. Ruppert. 1988. *Transformation and Weighting in Regression*. London: Chapman and Hall.

Chitty, L. S., D. G. Altman, A. Henderson, and S. Campbell. 1994. Charts of fetal size: 2. Head Measurements. *British Journal of Obstetrics and Gynaecology* 101: 35–43.

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

Royston P. and E. M. Wright. 1997a. How to construct "normal ranges" for fetal variables. *Ultrasound in Obstetrics and Gynecology* (submitted).

Royston, P. and E. M. Wright. 1997b. A parametric method for estimating age-specific reference intervals ("normal ranges"). *Journal of the Royal Statistical Society, Series A* (forthcoming).

Wright, E. and P. Royston. 1996. sbe13: Age-specific reference interval ("normal ranges"). *Stata Technical Bulletin* 34: 24–34. Reprinted in *Stata Technical Bulletin Reprints*, vol. 6, pp. 91–104.

sbe16 Meta-analysis

Stephen Sharp, London School of Hygiene and Tropical Medicine, London, stephen.sharp@lshtm.ac.uk
Jonathan Sterne, St Thomas' Hospital, London, j.sterne@umds.ac.uk

The command `meta` performs the statistical methods involved in a systematic review of a set of individual studies, reporting the results in text and also optionally in a graph. Each of the individual studies is a comparison of the effect on the study outcome of two exposure groups or, as is often the case in clinical trials, two treatment regimens.

Background

Given an estimate of treatment effect (for example a log odds ratio) and its standard error from a number of studies, the statistical methods used to combine the evidence across studies are well known (see Carlin 1992, for example), and are summarized below.

Suppose there are k studies, each with 2 comparison groups of subjects. Let θ_i denote the true treatment effect in trial i, $\hat{\theta}_i$ the estimated treatment effect in trial i, and v_i the variance of the estimated treatment effect.

Fixed-effects model

Under the assumption of a true treatment effect fixed across all studies, $\theta_i = \theta$, say, a minimum variance unbiased estimator of θ is

$$\hat\theta_F = \frac{\sum_{i=1}^k w_i\hat\theta_i}{\sum_{i=1}^k w_i}$$

where $w_i = 1/v_i$. The variance of $\hat\theta_F$ is $1/\sum_{i=1}^k w_i$.

Test for heterogeneity across studies

A test of the hypothesis $\theta_i = \theta$ for all i is a test for true differences between trials (i.e., heterogeneity). Under the null hypothesis, the statistic $Q = \sum_{i=1}^k w_i(\hat\theta_i - \hat\theta)^2$ has a χ^2_{k-1} distribution.

Random-effects model

One model to "allow" for heterogeneity between studies is $\theta_i \sim N(\theta, \tau^2)$. The most commonly used estimator of the between studies variance τ^2 is a moment estimator put forward by DerSimonian and Laird (1986):

$$\hat\tau^2 = \max\left[0, \frac{Q-(k-1)}{\sum_{i=1}^k w_i - \left(\frac{\sum_{i=1}^k w_i^2}{\sum_{i=1}^k w_i}\right)}\right]$$

An overall random-effects estimate can then be calculated as

$$\hat\theta_R = \frac{\sum_{i=1}^k w_i^*\theta_i}{\sum_{i=1}^k w_i^*}$$

where $w_i^* = 1/(v_i + \hat\tau^2)$. The variance of $\hat\theta_R$ is $1/\sum_{i=1}^k w_i^*$.

The heterogeneity between studies is reflected by an estimate $\hat\theta_R$ which is less precise (i.e., has greater variance) than the corresponding estimate assuming no heterogeneity $\hat\theta_F$.

Empirical Bayes estimates for each study

If the estimated between studies variance $\hat\tau^2$ is nonzero, empirical Bayes estimates can be calculated for each study:

$$\text{ebest}_i = \frac{\frac{\hat\theta_i}{v_i} + \frac{\hat\theta_R}{\hat\tau^2}}{\frac{1}{v_i} + \frac{1}{\hat\tau^2}}$$

Empirical Bayes estimates are shrunk towards the overall random effects estimate by a factor which depends on the relative magnitude of the estimated within and between study variances.

The variance of ebest_i is

$$\frac{\hat\tau^2 v_i}{\hat\tau^2 + v_i} + \frac{\left(\frac{v_i}{\hat\tau^2 + v_i}\right)^2}{\sum_{i=1}^k w_i^*}$$

Syntax

The command `meta` works on a dataset containing the estimated effect *theta* and its standard error *setheta* for each study. The syntax is

```
meta theta setheta [if exp] [in range] [, eform print ebayes level(#) graph(f|r|e)
    id(strvar) fmult(#) boxysca(#) boxshad(#) cline ltrunc(#) rtrunc(#) graph_options ]
```

By default, the output from the command contains the pooled estimate, lower and upper confidence limits, and test of the null hypothesis that the true pooled effect is 0, for each of the fixed- and random-effects models. The result of the χ^2 test of no true differences between the study effects (no heterogeneity) and the DerSimonian and Laird estimator of between studies variance are also reported.

Options for displaying results

`eform` specifies that all output, both default and estimates on the optional graph or in the optional print-out, are presented on an exponential scale (i.e., the original estimates are exponentiated). If the `ebayes` option is invoked, the variable `ebest` is also on an exponential scale. This option is useful where the original estimates of effect are on a log scale, such as a log odds ratio or log rate ratio.

`print` provides a listing of the weights used in the fixed- and random-effects estimation, together with individual estimates and confidence intervals for each study. The individual study estimates are calculated from the raw data by default, or are empirical Bayes estimates if the `ebayes` option is invoked.

`ebayes` creates two new variables in the dataset: `ebest` contains empirical Bayes estimates for each study, and `ebse` the corresponding standard errors. Any existing variables called `ebest` or `ebse` in the dataset are overwritten.

`level(`*#*`)` gives the level for the confidence limits (default 95).

Options for graphing results

`graph(f|r|e)` produces a graph showing the estimates and confidence intervals for each study, together with the combined estimate and confidence interval from the fixed-effects model if `f` is specified, and from the random-effects model if `r` is specified. The estimates are plotted with boxes; the area of each box is inversely proportional to the estimated effect's variance in that study, hence giving more visual prominence to studies where the effect is more precisely estimated. If `e` is specified, the empirical Bayes estimates from each study are plotted, together with the combined estimate from the random-effects model, and in this case the options `print` and `ebayes` are automatically invoked.

`id(`*strvar*`)` supplies a string variable which is used to label the studies on the graph, and, if the `print` option is invoked, in the listing of individual weights and study estimates.

`fmult(`*#*`)` is a number greater than zero which can be used to scale the font size for the study labels. The font size is automatically reduced if the maximum label length is greater than 8, or the number of studies is greater than 20. However it may be possible to increase it somewhat over the default size.

`boxysca(`*#*`)` provides a number between 0 and 1 which can be used to reduce the vertical length of the boxes. This is used to make boxes square if a vertical magnification of more than 100 has been used to increase the length of the graph. The default is 1.

`boxshad(`*#*`)` provides an integer between 0 and 4 which gives the box shading (0 most, 4 no shading). The default is 0.

`cline` asks that a vertical dotted line be drawn through the combined estimate.

`ltrunc(`*#*`)` truncates the left side of the graph at the number *#*. This is used to truncate very wide confidence intervals. However, *#* must be less than each of the individual study estimates.

`rtrunc(`*#*`)` truncates the right side of the graph at the number *#*, and must be greater than each of the individual study estimates.

graph_options are any options allowed with `graph, twoway` other than `ylabel()`, `symbol()`, `xlog`, `ytick`, and `gap`.

Example

Pre-eclampsia is a serious condition which can develop in the second half of pregnancy, affecting in total about 7% of pregnancies. Untreated, it can lead to eclampsia, which may result in maternal or fetal death.

We illustrate the use of `meta` with data from 9 randomized clinical trials of the use of diuretics for various manifestations of pre-eclampsia in pregnancy. An overview of these data was published by Collins, Yusef, and Peto (1985), and we focus on the effect of diuretics on the risk of any pre-eclampsia.

```
. use diuretic, clear
(Diuretics and pre-eclampsia)

. describe

Contains data from diuretic.dta
  obs:             9                             Diuretics and pre-eclampsia
 vars:             6
 size:           189
```

```
-------------------------------------------------------------------------------
   1. trial       byte   %9.0g       trlab      trial identity number
   2. trialid     str8   %9s                    trial first author
   3. nt          int    %9.0g                  total treated patients
   4. nc          int    %9.0g                  total control patients
   5. rt          int    %9.0g                  pre-eclampsia treated
   6. rc          int    %9.0g                  pre-eclampsia control
-------------------------------------------------------------------------------

. list, noobs

    trial    trialid         nt         nc         rt         rc
        1    Weseley        131        136         14         14
        2    Flowers        385        134         21         17
        3    Menzies         57         48         14         24
        4     Fallis         38         40          6         18
        5    Cuadros       1011        760         12         35
        6   Landesma       1370       1336        138        175
        7      Kraus        506        524         15         20
        8    Tervila        108        103          6          2
        9   Campbell        153        102         65         40
```

Before `meta` can be used, it is necessary first to calculate the estimated effect, which in this case will be the log odds ratio, and its standard error, for each study.

```
. gen logor=log((rt/(nt-rt))/(rc/(nc-rc)))

. gen selogor=sqrt((1/rc)+(1/(nc-rc))+(1/rt)+(1/(nt-rt)))

. meta logor selogor, eform

Meta-analysis of 9 studies (exponential form)
-----------------------------------------------

Fixed and random effects pooled estimates,
lower and upper 95% confidence limits, and
asymptotic z-test for null hypothesis that true effect=0

Fixed effects estimation
   Est   Lower   Upper  z_value  p_value
 0.672   0.564   0.800  -4.455   0.000

Test for heterogeneity: Q= 27.265 on 8 degrees of freedom (p= 0.001)
Der Simonian and Laird estimate of between studies variance =  0.230

Random effects estimation
   Est   Lower   Upper  z_value  p_value
 0.596   0.400   0.889  -2.537   0.011
```

The output, on an odds scale, shows that there is strong evidence of heterogeneity between the 9 trials, and taking into account the additional variability between studies in a random-effects model, the odds ratio of pre-eclampsia comparing diuretics with placebo is 0.596 (with 95% confidence interval 0.400 to 0.889).

The `graph(r)` option may be used to produce a graph showing the combined random-effects estimate.

```
. meta logor selogor, eform graph(r) id(trialid) cline xlab(0.5,1,1.5) xline(1)
> boxsh(4) b2("Odds ratio - log scale")
```

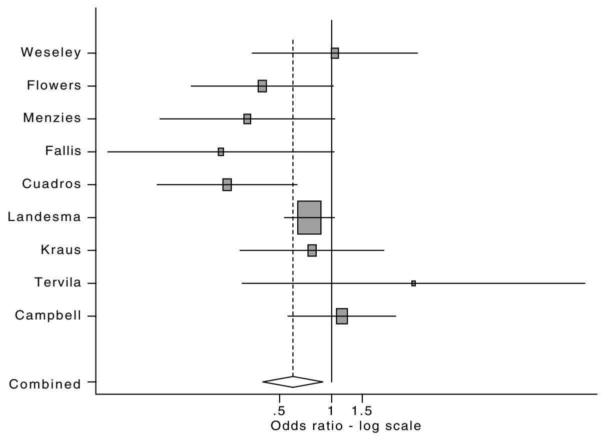

Figure 1.

Alternatively, the `graph(e)` option may be used to plot the empirical Bayes estimates, and a combined random-effects estimate. This also invokes automatically the `print` and `ebayes` options, hence listing the individual study weights and empirical Bayes estimates, as well as creating variables `ebest` and `ebse` in the dataset.

```
. meta logor selogor, eform graph(e) id(trialid) cline xlabel(0.5,1,1.5) xline(1)
> boxsh(2) b2("Odds ratio - log scale")
Meta-analysis of 9 studies (exponential form)
-----------------------------------------------
Fixed and random effects pooled estimates,
lower and upper 95% confidence limits, and
asymptotic z-test for null hypothesis that true effect=0
Fixed effects estimation
   Est   Lower   Upper  z_value  p_value
 0.672   0.564   0.800  -4.455   0.000
Test for heterogeneity: Q= 27.265 on 8 degrees of freedom (p= 0.001)
Der Simonian and Laird estimate of between studies variance =  0.230
Random effects estimation
   Est   Lower   Upper  z_value  p_value
 0.596   0.400   0.889  -2.537   0.011
Weights given to each study in fixed and random effects estimation,
estimates of effect in each study,
and lower and upper 95% confidence limits
Note: estimates and confidence limits are empirical Bayes
       Study   Fixed    Rand     Est   Lower   Upper
     Weseley    6.27    2.57    0.83    0.44    1.55
     Flowers    8.49    2.88    0.46    0.26    0.80
     Menzies    5.62    2.45    0.42    0.22    0.81
      Fallis    3.35    1.89    0.39    0.19    0.83
     Cuadros    8.75    2.91    0.33    0.19    0.58
    Landesma   68.34    4.09    0.73    0.58    0.92
       Kraus    8.29    2.85    0.71    0.40    1.24
     Tervila    1.46    1.09    0.89    0.38    2.12
    Campbell   14.73    3.36    0.99    0.62    1.56
```

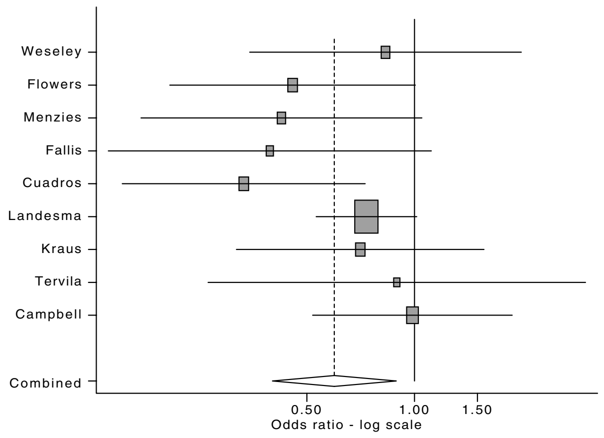

Figure 2.

```
. describe
Contains data from diur.dta
  obs:            9                          Diuretics and pre-eclampsia
 vars:           10                          13 May 1997 09:10
 size:          333 (99.9% of memory free)
-------------------------------------------------------------------------------
   1. trial       byte   %9.0g       trlab      trial identity number
   2. trialid     str8   %9s                    trial first author
   3. nt          int    %9.0g                  total treated patients
   4. nc          int    %9.0g                  total control patients
   5. rt          int    %9.0g                  pre-eclampsia treated
   6. rc          int    %9.0g                  pre-eclampsia control
   7. logor       float  %9.0g
   8. selogor     float  %9.0g
```

```
      9. ebest       float  %9.0g
     10. ebse        float  %9.0g
   ---------------------------------------------------------------------------
```

Individual or frequency records

meta operates on data contained in frequency records, one record per study, as was the case in the example, and will be the case with any data taken from published papers, often the source of data for a meta-analysis. If the data are in individual records, one record per subject with a variable indicating to which study the subject belongs, as would be the case in an individual patient data (IPD) meta-analysis, the records must first be combined into frequency records before meta can be used. Stata commands such as collapse and byvar will be appropriate for this manipulation; an example appears in the on-line help for meta.

Saved results

meta saves the following results in the S_ macros:

S_1	$\hat\theta_F$, combined fixed-effects estimate
S_2	SE of $\hat\theta_F$
S_3	Lower confidence limit on $\hat\theta_F$
S_4	Upper confidence limit on $\hat\theta_F$
S_5	Z-statistic to test null hypothesis that $\theta_F = 0$
S_6	p-value for test of null hypothesis that $\theta_F = 0$
S_7	$\hat\theta_R$, combined random-effects estimate
S_8	SE of $\hat\theta_R$
S_9	Lower confidence limit on $\hat\theta_R$
S_10	Upper confidence limit on $\hat\theta_R$
S_11	Z-statistic to test null hypothesis that $\theta_R = 0$
S_12	p-value for test of null hypothesis that $\theta_R = 0$
S_13	$\hat\tau^2$, estimate of between studies variance

Acknowledgment

We are grateful to Ian White for a variety of suggestions, including the incorporation of empirical Bayes estimates into the program.

References

Carlin, J. B. 1992. Meta-analysis for 2 × 2 tables: A Bayesian approach. *Statistics in Medicine* 11: 141–158.

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

Collins R., S. Yusuf, and R. Peto. 1985. Overview of randomised trials of diuretics in pregnancy. *British Medical Journal* 290: 17–23.

sg70 Interquantile and simultaneous-quantile regression

William Gould, StataCorp, wgould@stata.com

Linear regression measures $E(y|\mathbf{x}) = \mathbf{xb}$. Quantile regression focuses on the quantiles instead of the expected value and measures $\mathrm{Q}_\alpha(y|\mathbf{x}) = \mathbf{xb}_\alpha$. For instance, $\mathrm{Q}_{.50}(y|\mathbf{x}) = \mathbf{xb}_{.50}$ reflects the median of y given $\mathbf{x}$. If the distribution of $y|\mathbf{x}$ is symmetric, the mean is equal to the median and both estimators will asymptotically converge to the same limiting value.

A unique feature of quantile regression is its ability to estimate parameters appropriate for quantiles other than the median. For instance, one can obtain $\mathrm{Q}_{.25}(y|\mathbf{x}) = \mathbf{xb}_{.25}$ reflecting the lower quartile of the data or $\mathrm{Q}_{.75}(y|\mathbf{x}) = \mathbf{xb}_{.75}$ reflecting the upper quartile. If the distribution of $y|\mathbf{x}$ has constant variance, then the $\mathrm{Q}_{.25}$ and the $\mathrm{Q}_{.75}$ relationship will simply parallel the $\mathrm{Q}_{.50}$ relationship in that all estimated coefficients except for the intercepts will be roughly the same. If the coefficients differ, this is evidence of heteroskedasticity.

Heteroskedasticty—changing variance—divergent quantiles—say it how you will—can itself be of substantive interest. Say I tell you that treatment A and treatment B both lower blood pressure by roughly the same amount. Treatment A, however, is very consistent about the lowering. Treatment B is inconsistent, sometimes lowering blood pressure a lot, sometimes a little, but with roughly the same median (or expected) lowering as treatment A. Such would be suggested if we obtained the estimates

$$\mathrm{Q}_{.25}(y|\mathbf{x}) = 100 - 10 \cdot \mathrm{A} - 20 \cdot \mathrm{B}$$
$$\mathrm{Q}_{.50}(y|\mathbf{x}) = 160 - 10 \cdot \mathrm{A} - 10 \cdot \mathrm{B}$$
$$\mathrm{Q}_{.75}(y|\mathbf{x}) = 220 - 10 \cdot \mathrm{A} - 4 \cdot \mathrm{B}$$

Say I tell you that being black in America is associated with lower income, other things held constant. The policy implications are different if the distribution of income is unaffected by being black except for the shift rather than the distribution being more or less skewed around the lower mean or median. If special programs are to exist, should they be aimed at all blacks equally, the poorest blacks, or the richest blacks?

These and other questions like them can be addressed by comparing estimates for various quantiles. There is, however, a statistical difficulty. Standard quantile regression provides no estimate for the variances of the differences in the coefficients of separately estimated quantile regressions. Such estimates can be obtained by bootstrapping. The two commands described below provide such bootstrap estimates.

Syntax

`iqreg` *depvar* [*varlist*] [`if` *exp*] [`in` *range*] [, `quantiles(`*# #*`)` `reps(`*#*`)` `nolog` `level(`*#*`)`]

`sqreg` *depvar* [*varlist*] [`if` *exp*] [`in` *range*] [, `quantiles(`*#* [*#* [*#* ...]]`)` `reps(`*#*`)` `nolog` `level(`*#*`)`]

These commands share the features of all estimation commands.

To reset problem-size limits, see [R] **matsize**. Due to how `iqreg` is implemented, no more than 336 independent variables may be specified regardless of the value of `matsize`. Due to how `sqreg` is implemented, no more than 336 coefficients may be simultaneously estimated. This means no more than $336/q$ variables where q is the number of `quantiles()` specified. For instance, if 2 quantiles are specified, no more than 168 independent variables may be specified; if 3 quantiles are specified, no more than 112 independent variables may be specified; and so on.

Description

`iqreg` estimates interquantile range regressions, regressions of the difference in quantiles. If the `quantile()` option is not specified, the default is the interquartile range. The estimated variance-covariance matrix of the estimators (VCE) is obtained via bootstrapping.

`sqreg` estimates quantile regressions. It produces the same coefficients as `qreg` would produce were each quantile estimated separately. Reported standard errors will be similar, the difference being that `sqreg` obtains an estimate of the VCE via bootstrapping rather than the analytical formula. (In this sense, `sqreg` is similar to `bsqreg`, the bootstrapped variation of `qreg`.)

`sqreg` differs from `qreg` (and `bsqreg`) in that it can estimate results for multiple quantiles simultaneously, meaning that the calculated VCE includes between-quantiles blocks. Thus, one can test and construct confidence intervals comparing coefficients describing different quantiles.

Options

`quantiles(`*# #*`)` (the `quantiles()` option for the `iqreg` command) specifies the quantiles to be compared. Not specifying this option is equivalent to specifying `quantiles(.25 .75)`, meaning the interquartile range. Specifying `quantiles(.1 .9)` would estimate a model of the difference in the .9 and .1 quantiles.

If this option is specified, the first number must be less than the second.

Strictly speaking, both numbers should be between 0 and 1, exclusive. However, if you specify a number larger than 1, it will be interpreted as a percent. Thus, `quantiles(.25 .75)` could also be specified as `quantiles(25 75)`.

You may optionally place a comma between the two numbers. The default could be specified `quantiles(.25,.75)` or `quantiles(25,75)`.

`quantiles(`*#* [*#* [*#* ...]]`)` (the `quantiles()` option for the `sqreg` command) specifies the quantiles to be estimated. For instance, `quantiles(.25 .75)` specifies that two equations are to be estimated, one for the .25 quantile and another for the .75. `quantiles(.25 .50 .75)` specifies three equations; a .50 quantile (median) regression is to be added.

Strictly speaking, numbers should be between 0 and 1, exclusive. However, if you specify a number larger than 1, it will be interpreted as a percent. Thus, `quantiles(.25 .50 .75)` could also be specified as `quantiles(25 50 75)` or even `quantiles(25 .5 75)`.

You may optionally place a comma between the two numbers. You may type `quantiles(25 50 75)` or `quantiles(25,50,75)`.

Quantiles may be specified in any order. Results will be more easily read if you specify them in ascending order.

`reps(#)` specifies the number of bootstrap replications to be used to obtain an estimate of the variance-covariance matrix of the estimators (standard errors). `reps(20)` is the default.

This default is arguably too small. `reps(100)` would perform 100 bootstrap replications. `reps(1000)` would perform 1,000.

`nolog` specifies intermediate output during the estimation process is not to be presented. If `nolog` is not specified, a period is placed on the screen after the completion of each replication (so if `reps(100)` is specified, 100 periods appear before final results are presented). `nolog` suppresses this.

`level(#)` specifies the confidence level in percent for the confidence interval of the coefficients.

Remarks

If you are not familiar with quantile regression, please see [R] **qreg**.

Consider a quantile-regression model where the qth quantile is given by

$$Q_q(y) = a_q + b_{q,1}x_1 + b_{q,2}x_2$$

For instance, the 75th and 25th quantiles are given by

$$Q_{.75}(y) = a_{.75} + b_{.75,1}x_1 + b_{.75,2}x_2$$
$$Q_{.25}(y) = a_{.25} + b_{.25,1}x_1 + b_{.25,2}x_2$$

The difference in the quantiles is then

$$Q_{.75}(y) - Q_{.25}(y) = (a_{.75} - a_{.25}) + (b_{.75,1} - b_{.25,1})x_1 + (b_{.75,2} - b_{.25,2})x_2$$

`qreg` estimates models such as $Q_{.75}(y)$ and $Q_{.25}(y)$. `iqreg` estimates models such as $Q_{.75}(y) - Q_{.25}(y)$. The relationship of the coefficients estimated by `qreg` and `iqreg` are exactly as shown: `iqreg` reports coefficients that are the difference in coefficients of two `qreg` models and, of course, `iqreg` reports the appropriate standard errors which it obtains by bootstrapping.

The other new command, `sqreg`, is like `qreg` in that it estimates the equations for the quantiles

$$Q_{.75}(y) = a_{.75} + b_{.75,1}x_1 + b_{.75,2}x_2$$
$$Q_{.25}(y) = a_{.25} + b_{.25,1}x_1 + b_{.25,2}x_2$$

The coefficients it obtains are the same as would be obtained by estimating each equation separately using the existing `qreg`. `sqreg` differs from `qreg` in that it estimates the equations simultaneously and obtains an estimate of the entire variance-covariance matrix of the estimators by bootstrapping. Thus, one can perform hypothesis tests concerning coefficients both within and across equations.

For example, to obtain estimates of the above model, you could type

```
. qreg y x1 x2, q(.25)
. qreg y x1 x2, q(.75)
```

Doing this, you would obtain estimates of the parameters but you could not test whether $b_{.25,1} = b_{.75,1}$ or, equivalently $b_{.75,1} - b_{.25,1} = 0$. If your interest really is in the difference of coefficients, you could type

```
. iqreg y x1 x2, q(.25 .75)
```

The "coefficients" reported would be the difference in quantile coefficients. Alternatively, you could estimate both quantiles simultaneously and then test the equality of the coefficients:

```
. sqreg y x1 x2, q(.25 .75)
. test [q25]x1 = [q75]x2
```

Whether you use `iqreg` or `sqreg` makes no difference in terms of this test. `sqreg`, however, because it estimates the quantiles simultaneously, allows testing other hypotheses. `iqreg`, by focusing on quantile differences, presents results in a way that are easier to read.

Finally, `sqreg` can estimate quantiles singly,

```
. sqreg y x1 x2, q(.5)
```

or it can estimate multiple quantiles simultaneously,

```
. sqreg y x1 x2, q(.25 .5 .75)
```

Example

Using a 1988 sample of 2,377 working women, an economist estimates the following linear regression:

```
. regress ln_wage ed tenure
  Source |       SS       df       MS                  Number of obs =    2377
---------+------------------------------               F(  2,  2374) =  321.14
   Model |  182.195981     2  91.0979903               Prob > F      =  0.0000
Residual |  673.425626  2374  .283667071               R-squared     =  0.2129
---------+------------------------------               Adj R-squared =  0.2123
   Total |  855.621607  2376  .360110104               Root MSE      =   .5326
------------------------------------------------------------------------------
 ln_wage |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
      ed |   .0893355   .0044465     20.091   0.000       .0806161    .0980548
  tenure |   .0262681   .0019983     13.145   0.000       .0223495    .0301867
   _cons |   .5500002   .0595964      9.229   0.000       .4331338    .6668666
------------------------------------------------------------------------------
```

`ln_wage` refers to the log of the hourly wage, `ed` to years of schooling completed, and `tenure` to years on the current job. Economists often interpret coefficients of regressions of $\ln(y)$ on x as the proportional change in y for a unit change in x because $d\ln y/dx = (1/y)dy/dx$. Thus, an additional year of schooling is estimated to increase the wage by roughly 8.9% and an additional year of tenure by 2.6%.

The median wage given `ed` and `tenure` could be obtained by estimating a quantile regression:

```
. qreg ln_wage ed tenure, q(.5)
Iteration  1:  WLS sum of weighted deviations =  907.85532
Iteration  1: sum of abs. weighted deviations =  907.60506
Iteration  2: sum of abs. weighted deviations =  906.35635
(output omitted)
Iteration  8: sum of abs. weighted deviations =  904.22524
Median Regression                                    Number of obs =      2377
  Raw sum of deviations 1104.709 (about 1.8564485)
  Min sum of deviations 904.2252                     Pseudo R2     =    0.1815
------------------------------------------------------------------------------
 ln_wage |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
      ed |   .0947895   .0045872     20.664   0.000       .0857942    .1037849
  tenure |   .0305867   .0020605     14.844   0.000       .0265461    .0346273
   _cons |   .4268735    .061496      6.941   0.000       .3062821    .5474649
------------------------------------------------------------------------------
```

These results are similar to those produced by linear regression.

The researcher is also interested in the variation of wages, and to examine that, estimates models for the 25th and 75th percentiles:

```
. qreg ln_wage ed tenure, q(.25) nolog
.25 Quantile Regression                              Number of obs =      2377
  Raw sum of deviations 848.9257 (about 1.4788921)
  Min sum of deviations 712.3449                     Pseudo R2     =    0.1609
------------------------------------------------------------------------------
 ln_wage |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
      ed |   .0850433   .0048776     17.436   0.000       .0754786    .0946081
  tenure |   .0337707   .0024232     13.937   0.000       .0290189    .0385224
   _cons |   .2415063    .065603      3.681   0.000       .1128612    .3701514
------------------------------------------------------------------------------
```

```
. qreg ln_wage ed tenure, q(.75) nolog
.75 Quantile Regression                                Number of obs =      2377
  Raw sum of deviations 903.5743 (about 2.2698486)
  Min sum of deviations 772.0451                       Pseudo R2     =    0.1456
------------------------------------------------------------------------------
 ln_wage |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
      ed |   .0990176   .0052676     18.798   0.000        .088688    .1093471
  tenure |   .0243454   .0023039     10.567   0.000       .0198275    .0288633
   _cons |   .6931128   .0695066      9.972   0.000       .5568128    .8294128
------------------------------------------------------------------------------
```

In the above, we specified `qreg`'s `nolog` option to prevent displaying the iteration log and so saved some paper.

Note what the researcher found:

Variable	25th percentile	50th percentile	75th percentile
ed	.085	.094	.099
tenure	.034	.030	.024
intercept	.241	.427	.693

The distribution of log wages appears to spread out with increasing education (the effect of ed at the 25th percentile is less than at the 50th percentile which is less than the effect at the 75th percentile) and the spread of log wages appears to contract with increases in tenure.

All we can say, having estimated these equations separately, is that such a result appears in the data. We cannot be more precise because the estimates have been made separately. With `sqreg`, however, we can estimate all the effects simultaneously:

```
. sqreg ln_wage ed tenure, q(.25 .5 .75) rep(100)
(estimating base model)
(bootstrapping .........................(output omitted)...)
Simultaneous quantile Regression                       Number of obs =      2377
  bootstrap(100) SEs                                   .25 Pseudo R2 =    0.1609
                                                       .50 Pseudo R2 =    0.1815
                                                       .75 Pseudo R2 =    0.1456
------------------------------------------------------------------------------
         |             Bootstrap
 ln_wage |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
q25      |
      ed |   .0850433   .0044446     19.134   0.000       .0763276    .0937591
  tenure |   .0337707   .0023117     14.609   0.000       .0292375    .0383038
   _cons |   .2415063   .0563495      4.286   0.000        .131007    .3520056
---------+--------------------------------------------------------------------
q50      |
      ed |   .0947895   .0040758     23.256   0.000        .086797    .1027821
  tenure |   .0305867   .0019114     16.002   0.000       .0268384     .034335
   _cons |   .4268735   .0531761      8.028   0.000       .3225972    .5311498
---------+--------------------------------------------------------------------
q75      |
      ed |   .0990176    .005883     16.831   0.000       .0874813    .1105539
  tenure |   .0243454   .0020002     12.171   0.000        .020423    .0282678
   _cons |   .6931128   .0774286      8.952   0.000       .5412781    .8449475
------------------------------------------------------------------------------
```

The coefficient estimates above are the same as those previously estimated although the standard error estimates are a little different. `sqreg` obtains estimates of variance by bootstrapping. Rogers (1992) provides evidence that, in the case of quantile regression, the bootstrap standard errors are better than those calculated analytically by Stata.

The important thing here, however, is that the full covariance matrix of the estimators has been estimated and stored and thus it is now possible to perform hypotheses tests. Are the effects of education the same as the 25th and 75th percentiles?

```
. test [q25]ed = [q75]ed
 ( 1)  [q25]ed - [q75]ed = 0.0
       F(  1,  2374) =    5.20
            Prob > F =    0.0226
```

It appears that they are not. We can obtain a confidence interval for the difference using `lincom`:

```
. lincom [q75]ed-[q25]ed
 ( 1) - [q25]ed + [q75]ed = 0.0
------------------------------------------------------------------------------
 ln_wage |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
     (1) |   .0139742   .0061257      2.281   0.023       .0019619    .0259866
------------------------------------------------------------------------------
```

Indeed, we could test whether the full set of coefficients are equal at the three quantiles estimated:

```
. test [q25]ed = [q50]ed, notest
 ( 1)  [q25]ed - [q50]ed = 0.0
. test [q25]ed = [q75]ed, notest accum
 ( 1)  [q25]ed - [q50]ed = 0.0
 ( 2)  [q25]ed - [q75]ed = 0.0
. test [q25]tenure = [q50]tenure, notest accum
 ( 1)  [q25]ed - [q50]ed = 0.0
 ( 2)  [q25]ed - [q75]ed = 0.0
 ( 3)  [q25]tenure - [q50]tenure = 0.0
. test [q25]tenure = [q75]tenure, accum
 ( 1)  [q25]ed - [q50]ed = 0.0
 ( 2)  [q25]ed - [q75]ed = 0.0
 ( 3)  [q25]tenure - [q50]tenure = 0.0
 ( 4)  [q25]tenure - [q75]tenure = 0.0
       F(  4,  2374) =    5.33
            Prob > F =    0.0003
```

`iqreg` focuses on one quantile comparison but presents results that are more easily interpreted:

```
. iqreg ln_wage ed tenure, q(.25 .75) reps(100)
(estimating base model)
(bootstrapping .........................(output omitted)...)
.75-.25 Interquantile Regression                    Number of obs =      2377
  bootstrap(100) SEs                                .75 Pseudo R2 =    0.1456
                                                    .25 Pseudo R2 =    0.1609
------------------------------------------------------------------------------
         |              Bootstrap
 ln_wage |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
      ed |   .0139742    .005438      2.570   0.010       .0033105     .024638
  tenure |  -.0094252   .0024666     -3.821   0.000      -.0142622   -.0045883
   _cons |   .4516065   .0712686      6.337   0.000       .3118514    .5913616
------------------------------------------------------------------------------
```

The above output makes clear the nature of the dispersion in the data: Increases in education are associated with an increase in dispersion; increases in job tenure decrease dispersion.

If one took seriously the above results—the data is real but we have hardly done the work necessary to ensure that these results have any validity—one policy implication would be that, were income equality a goal, government should not subsidize education. Increasing education increases the dispersion of log wage (which is to say, drastically increases the dispersion of wages). On the other hand, these results also suggest that increasing education increases wages.

Increased job tenure, on the other hand, is associated with higher levels and lesser dispersion of wages. Perhaps that is just a quirk of this data.

I do not want to make too much of these results; the purpose of this example is simply to illustrate these two new commands and to do so in a context that suggests why analyzing dispersion might be of interest.

In terms of numeric results, note that `lincom` after `sqreg` produced a t statistic of 2.281 for the difference $b_{.75,\mathrm{ed}} - b_{.25,\mathrm{ed}}$ whereas `iqreg` above reported a t statistic of 2.570. The difference is due solely to the randomness of the bootstrap procedure. If we increased the number of replications, the results would converge. Mechanically, if you set the random number seed to the same value before estimation, and specify the same number of replications, results will be numerically identical.

Coverage

To verify that the standard errors are about right, I performed simulations on the model

$$y = 0 + 1x_1 + 2x_2 + \epsilon$$

where $\epsilon \sim N(0,1)$ and $\epsilon \sim N(0,(1x_1+.6)^2)$. Each simulation contained 1,000 replications. A replication amounted to drawing ϵ from the assumed distribution and then estimating

```
. iqreg y x1 x2, reps(...)
```

where bootstrap standard errors were obtained with `reps(20)` (the default) and `reps(100)`.

For example, the first simulation with $\epsilon \sim N(0,1)$ produced a dataset that, had I estimated a linear regression, would have produced

```
. reg y x1 x2
  Source |       SS       df       MS                  Number of obs =    1000
---------+------------------------------               F(  2,   997) =  211.12
   Model |  420.535523     2  210.267761               Prob > F      =  0.0000
Residual |  992.998133   997  .995986091               R-squared     =  0.2975
---------+------------------------------               Adj R-squared =  0.2961
   Total |  1413.53366   999   1.4149486               Root MSE      =  .99799

------------------------------------------------------------------------------
       y |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
      x1 |   .9620749   .1082926      8.884   0.000       .7495674    1.174582
      x2 |   1.999471   .1083726     18.450   0.000       1.786806    2.212135
   _cons |   .0387563   .0816323      0.475   0.635      -.1214345     .198947
------------------------------------------------------------------------------
```

and, correspondingly, would have produced the median regression

```
. qreg y x1 x2
Iteration  1:  WLS sum of weighted deviations =  797.29612

Iteration  1: sum of abs. weighted deviations =  797.29243
Iteration  2: sum of abs. weighted deviations =  797.23706
(output omitted)
Iteration 11: sum of abs. weighted deviations =  797.06048

Median Regression                                   Number of obs =      1000
  Raw sum of deviations 947.2333 (about 1.5966282)
  Min sum of deviations 797.0605                    Pseudo R2     =    0.1585

------------------------------------------------------------------------------
       y |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
      x1 |    .888463   .1149225      7.731   0.000       .6629452    1.113981
      x2 |   1.979701   .1149399     17.224   0.000       1.754149    2.205253
   _cons |   .0876677   .0867206      1.011   0.312      -.0825081    .2578434
------------------------------------------------------------------------------
```

The interest instead, however, was in the dispersion and I estimated

```
. iqreg y x1 x2
(estimating base model)
(bootstrapping ....................)

.75-.25 Interquantile Regression                    Number of obs =      1000
  bootstrap(20) SEs                                 .75 Pseudo R2 =    0.1460
                                                    .25 Pseudo R2 =    0.1788

------------------------------------------------------------------------------
         |             Bootstrap
       y |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
      x1 |  -.0623275   .1752463     -0.356   0.722      -.4062213    .2815664
      x2 |  -.4927516   .1263719     -3.899   0.000      -.7407369   -.2447662
   _cons |   1.634377   .1587514     10.295   0.000       1.322852    1.945902
------------------------------------------------------------------------------
```

These results correspond roughly to the true results. Since $\epsilon \sim N(0,1)$, the true results are $b_1 = 0$, $b_2 = 0$, and intercept $=$ 1.348979 (which is the difference in the 75th and 25th percentiles of the unit normal). In this case, the true value of each of the coefficients is contained in the 95% confidence interval. If we repeated this experiment 1,000 times, we would expect that each of the 95% confidence intervals would contain the true value in 95% of the experiments. That is, we would expect that to be true if the calculated standard errors are approximately correct. The actual percentage of confidence intervals containing the true value is called coverage and the results of repeating the experiment 1,000 times are

coefficient	true value	average value	average width	95% coverage	CI for coverage
`reps(20)`					
b1	0	0.0042	0.6928	94.4	92.8–95.7
b2	0	-.0059	0.6972	94.9	93.3–96.2
intercept	1.3490	1.3498	0.5212	93.4	91.8–94.9
`reps(100)`					
b1	0	-.0032	0.7011	96.8	95.5–97.8
b2	0	0.0063	0.7010	95.9	94.5–97.0
intercept	1.3490	1.3453	0.5266	96.0	94.6–97.1

In the table above, “average” refers to the average value of the estimated coefficient over the 1,000 experiments and “average width” refers to the average width of the reported 95% confidence interval.

The last column reports the 95% confidence interval for the observed coverage. For instance, in the `reps(20)` case the observed 95% coverage for b_1 was 94.4%, meaning 944 out of 1,000 reported confidence intervals contained the true value of 0. The 95% confidence interval for a binomial experiment with $k = 944$ and $n = 1{,}000$ is 92.8 to 95.7 percent.

Repeating the experiments for $\epsilon \sim N(0, (1x_1 + .6)^2)$, the true value of the coefficients are intercept $=$ .8058132, $b_1 = 1.348979$, and $b_2 = 0$:

coefficient	true value	average value	average width	95% coverage	CI for coverage
`reps(20)`					
b1	1.3490	1.3516	0.7486	93.4	91.7–94.9
b2	0	-.0057	0.7063	95.0	93.5–96.3
intercept	0.8094	0.8112	0.4746	94.3	92.7–95.7
`reps(100)`					
b1	1.3490	1.3428	0.7557	97.7	96.6–98.5
b2	0	0.0087	0.7099	95.4	93.9–96.6
intercept	0.8094	0.8058	0.4764	95.3	93.8–96.7

Performance

Estimating bootstrapped standard errors is never quick but those who have used `bsqreg` will be pleasantly surprised. In fact, even when your interest is not in comparing quantiles, you will want to use `sqreg` as an alternative to `bsqreg`. `sqreg` will estimate just one quantile if you wish.

Logically speaking, total estimation time should be linear in the number of bootstrap replications and, for the new `sqreg` and `iqreg`, it is. The older `bsqreg` had run times that were quadratic in number of replications and the effect of replications squared was larger the more observations in the dataset.

How much of an improvement you will notice depends on whether you use Windows, Macintosh, or Unix; Unix users will notice the least improvement except on large problems because, when datasets were small, the Unix file buffering system did a good job of covering for the shortcomings of `bsqreg`.

In any case, here are timings for a small dataset:

	60 MHz Pentium		120 MHz Pentium	
	Windows 95		Unix	
Replications	bsqreg	sqreg	bsqreg	sqreg
20	2.55	2.16	1.02	0.96
50	6.01	5.00	2.51	2.26
100	11.89	9.80	4.89	4.44
250	30.09	24.00	12.47	10.92
500	62.11	47.66	25.78	21.68
1000	132.62	95.00	55.03	43.29

All times are reported in seconds. The commands executed were

```
bsqreg:  bsqreg mpg weight displ foreign
 sqreg:  sqreg mpg weight displ for, q(.5)
```

with the automobile data loaded into memory.

References

Gould, W. W. 1992. sg11.1: Quantile regression with bootstrapped standard errors. *Stata Technical Bulletin* 9: 19–21. Reprinted in *Stata Technical Bulletin Reprints*, vol. 2, pp. 137–139.

Rogers, W. H. 1992. sg11. Quantile regression standard errors. *Stata Technical Bulletin* 9: 16–18. Reprinted in *Stata Technical Bulletin Reprints*, vol. 2, pp. 133–137.

sg71 Routines to maximize a function

Christopher Ferrall, Queen's University, Kingston, Ontario, Canada, FAX 613-545-6668, ferrallc@post.queensu.ca

amoeba and quasi are routines to maximize a multi-dimensional function. Both are based on translations of the Pascal code presented in Press et al. (1987). Some of the global options in the original code have been fixed as constants within the Stata code. One extension to quasi for use in likelihood maximization has been added, namely the bhhh option for computing the Hessian matrix. Readers interested in the technical details of the algorithms are referred to the lucid explanations in Press et al.

These routines can be used to perform maximum likelihood estimation, and the basic form of the call to the user-written objective function, *obj xin yout* is the same as the deriv0 form for ml. But amoeba and quasi do not incorporate many of the features of the built-in Stata ml command, nor do they require as much set up as ml. Instead, these routines are designed to be simple to use general-purpose optimization routines.

amoeba is an efficient search (non-derivative) algorithm for optimizing a multi-dimensional function developed by Nelder and Mead. Press et al. use the name amoeba because the algorithm moves a simplex of points in N dimensional space in a way that is very reminiscent of microbe locomotion. The algorithm is also coded and described by Barr in STB-32 (sg56) under the name simplex. A brief comparison to simplex is performed below.

By any name, the Nelder-Mead algorithm is more robust and effective than a simple grid search, and it works very well for any continuous (including non-differentiable) function. It can also make progress for discontinuous functions as well. It is a common problem to optimize a function that is well behaved (concave) only in a neighborhood of the optimal values. In this situation it is very effective to start out using amoeba and then pass on the results to a derivative-based routine (such as quasi) to complete convergence.

quasi implements the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton algorithm also described in Press et al. It uses numerical first derivatives, begins with the inverse Hessian set to the identity matrix, and by default updates the inverse Hessian with the BFGS step. The bhhh option specifies that the inverse Hessian instead be updated using the inverse of the outer-product of the gradient matrix (the Berndt–Hall–Hall–Hausman estimator). This requires that the objective return in the third argument the contribution to the likelihood function for each observation. Hence when using the bhhh option the call to the user-written objective function must take the form *obj x y objvar*, where *objvar* is a variable name.

Syntax

amoeba [*obj xin yout xout* [*stepsize itmax toler*]]

quasi [*obj xin yout xout* [*g h stepsize itmax toler* bhhh]]

where *obj* is the name of a program written by the user to evaluate the objective function to be maximized, *xin* is a row vector containing starting values, *yout* is a scalar to receive the maximum value of the objective function, and *xout* is a row vector to receive the value of the vector that maximizes the objective function.

Unless using the bhhh option under quasi, a call to *obj* must be of the form *obj x y* where *x* is a row vector at which the function is to be evaluated, and *y* is a scalar to receive the value of the function. When using the bhhh option the call to the user-written objective function must take the form *obj x y objvar*, where *objvar* is a variable name (see above).

Options

Under `amoeba`, *stepsize* is the percentage change in each parameter used to set up a simplex in the parameter space. Under `quasi`, it is the percentage step taken to compute the numerical gradient.

itmax is the maximum number of iterative steps that should be done.

toler is how "tight" the simplex (`amoeba`) or how small the gradients (`quasi`) must be before the algorithm quits.

g is a row vector to receive the final gradient in `quasi`.

h is a matrix to receive final inverse of hessian in `quasi`.

`bhhh` tells `quasi` to use the BHHH algorithm rather than BFGS in `quasi`.

Note that options are ordered and a period can be used to skip optional arguments. In addition, invoking `amoeba` or `quasi` with no arguments will display information about them, à la Unix.

Example 1: Ordinary least squares the hard way

Consider the linear regression

$$y = X\beta + \epsilon$$

where y is an $(N \times 1)$ vector of observed values, X is an $(N \times k)$ matrix of observed explanatory values, β is a $(k \times 1)$ vector of unknown parameters, and ϵ is the $(N \times 1)$ vector of disturbance terms. The ordinary least squares (OLS) estimate b of β solves:

$$b = \text{argmin}(y - X\beta)'(y - X\beta)$$

Of course, the solution is $b = (X'X)^{-1}X'y$. However, OLS is used here to illustrate `amoeba` and `quasi` by minimizing the sum of squared errors directly. We verify the results by comparing them to the output of Stata's `regress` command.

```
. set obs 100
. gen X = (_n-50)/10
. gen y = -2.0 + 3.0 * X + 5*invnorm(uniform())
. regress y X
  Source |       SS       df       MS                  Number of obs =     100
---------+------------------------------               F(  1,    98) =  364.75
   Model |  7699.47768     1  7699.47768               Prob > F      =  0.0000
Residual |  2068.67513    98  21.1089299               R-squared     =  0.7882
---------+------------------------------               Adj R-squared =  0.7861
   Total |  9768.15281    99  98.6682102               Root MSE      =  4.5944
------------------------------------------------------------------------------
       y |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
       X |   3.039786   .1591642     19.098   0.000        2.72393    3.355642
   _cons |   -2.34628   .4595135     -5.106   0.000       -3.25817   -1.434391
------------------------------------------------------------------------------
```

We code the objective function as the program `myols`.

```
program define myols
        tempvar e
        if "`3'"!="" {
                local sse = "`3'"
                qui capture drop `sse'
        }
        else {
                tempvar sse
        }
        matrix score double `e' = `1'
        qui replace `e' = ($yvar - `e')*($yvar - `e')
        qui egen double `sse' = sum(`e')
        scalar `2' = -`sse'
                * the minus because amoeba MAXimizes
end
```

Next we minimize the sum of squares using `amoeba`:

```
. global nlist = "beta:X beta:_cons"
. global yvar = "y"
. matrix b0 = (.1,5.1)
. matrix colnames b0 = $nlist
. amoeba myols b0 yout bols
-------------------- Starting Amoeba --------------------
__000001[1,2]:  Starting values
       beta:    beta:
          X    _cons
r1  0.10000  5.10000
Starting value of myols: -14597.87003
....................................
    Value of myols           Simplex Size        Iterations
    -2068.88961               9.6e-06             38
bols[1,2]:  Amoeba final values stored in bols
        beta:     beta:
           X     _cons
r1   3.03553  -2.30141
-------------------- Ending Amoeba--------------------
```

Notice that `amoeba` (and `quasi`) retain the column names of the starting vector `b0`, making it easier to calculate score matrices. `amoeba` reports that the SSE equals 14597.87003 at the initial guess `b0`. After each iteration `amoeba` and `quasi` print a period. After 38 iterations it convergences at the default criterion with SSE reduced to 2068.88961. This is less than 1% different than the OLS estimates reported in the regression table. The coefficients themselves have about the same precision. The final values are stored in the vector `bols`, which can be sent to `amoeba` again or to `quasi`:

```
. quasi myols bols yout bfinal g h
-------------------- Starting Quasi --------------------
bfinal[1,2]:  Starting values
        beta:     beta:
           X     _cons
r1   3.03553  -2.30141
Starting value of myols: -2068.88961
...
    Value of myols           Gradient Size       Iterations
    -2068.67513               1.8e-11              3
bfinal[1,2]:  Quasi final values stored in bfinal
        beta:     beta:
           X     _cons
c1   3.03979  -2.34628
-------------------- Ending Quasi --------------------
```

With good starting values and a quadratic objective function, `quasi` was able to converge almost exactly to the analytical solution in only three iterations of the algorithm.

Example 2: Powell's estimator for censored regression

This example is used to illustrate the use of `amoeba` on a non-differentiable objective function and to compare with Barr's `simplex` implementation of the algorithm. Using the same data as in Example 1, we now censor the y values:

```
. gen ycn = max(y,0)
```

The result is a classic tobit model, which can be estimated with maximum likelihood or through minimization of the non-smooth objective $|\texttt{ycn} - \max(X\beta, 0)|$. First, consider the `tobit` estimates:

```
. tobit ycn X, ll(0)
Tobit Estimates                                       Number of obs =    100
                                                      chi2(1)       = 107.27
                                                      Prob > chi2   = 0.0000
Log Likelihood = -137.38223                           Pseudo R2     = 0.2808
------------------------------------------------------------------------------
     ycn |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
       X |   3.154331   .3580773      8.809   0.000       2.443828    3.864834
   _cons |    -2.6174   1.048684     -2.496   0.014      -4.698217   -.5365839
---------+--------------------------------------------------------------------
     _se |   4.650711   .5214194             (Ancillary parameter)
------------------------------------------------------------------------------
Obs. summary:        58 left-censored observations at ycn<=0
                     42 uncensored observations
```

We can modify the last few lines of the `myols` program to compute the tobit likelihood value (put in a file called `tobin.ado`):

```
gen double `tob´ = normprob(-`xb´/`sig´) if $yvar<=0
replace `tob´ = exp( -0.5*(($yvar-`xb´)/`sig´)^2 )/(sqrt(2*_pi)*`sig´) if $yvar>0
replace `tob´ = ln(max(`tob´,1E-20))
gen double `yh´ = sum(`tob´)
scalar `2´ = `yh´[_N]
```

Starting from the same vector `b0` used with `myols`, and the intial value of `sig` = 3.0, `amoeba` reached a likelihood of -137.39104 after 86 iterations. These estimates can be polished using `quasi`, and since `tobin` is a proper likelihood function we can use the BHHH estimator for the Hessian, and ultimately the variance matrix for our estimates.

```
. global yvar = "ycn"
. matrix sig = (3.0)
. matrix btob = b0 , sig
. qui amoeba tobin btob yout bq1 . . 1E-5
. quasi tobin bq1 yout bfinal g h . . . bhhh
-------------------- Starting Quasi --------------------
bfinal[1,3]:  Starting values
        beta:     beta:
           X     _cons        c1
r1   3.15510  -2.70594   4.67081
Starting value of tobin: -137.39104
....
    Value of tobin           Gradient Size       Iterations
    -137.38223                2.1e-07              4
bfinal[1,3]:  Quasi final values stored in bfinal
              beta:     beta:
                 X     _cons        c1
__001F7Y   3.15433  -2.61741   4.65071
-------------------- Ending Quasi --------------------

. mat l h

symmetric h[3,3]
            __001FCP    __001FCY    __001FD7
__001FCP   .03100972
__001FCY  -.25118477   1.0938838
__001FD7   .16583154  -.33998424   .22516939
```

After only four iterations `quasi` converges. The inverse of the outer product of the gradient (returned in the matrix `h`) is a consistent estimate of the variance matrix. We can see that the square root of the diagonal of `h` is similar to the estimated standard errors reported by the built-in `tobit` command.

Now consider estimating the model nonparametrically. An ado program called `powell` is written which once again replaces the last few lines of `myols` with the new objective function:

```
qui replace `e´ = abs($yvar - max(`e´,0))
qui egen `sad´ = sum(`e´)
scalar `2´ = -`sad´
       * the minus because amoeba MAXimizes
global fvals = $fvals + 1
```

The global variable `fvals` is used to count the number of function evaluations performed during the estimation. To have a sense of what we should expect, we can compute the value of the objective function for the `tobit` estimates:

```
. predict xb, index
. egen tad = sum(abs(ycn-max(xb,0)))
. di tad
157.43954
```

The tobit model (which is correct given the data generating process) results in a value of 157.43954 for Powell's objective function. Since `powell` is not smooth we won't use `quasi` on it, so we will tighten up the convergence criterion on `amoeba` and display the results:

```
. amoeba powell b0 yout bpow1 . . 1E-7
-------------------- Starting Amoeba --------------------
__001FDH[1,2]:  Starting values
       beta:    beta:
          X    _cons
r1  0.10000  5.10000
Starting value of powell: -442.86853
..............................................................................
> ......
    Value of powell          Simplex Size        Iterations
    -154.70178                4.9e-08             85
bpow1[1,2]:  Amoeba final values stored in bpow1
        beta:     beta:
           X     _cons
r1   2.62174  -1.83524
-------------------- Ending Amoeba--------------------
. di "Number of function evaluations = " $fvals
Number of function evaluations = 160
```

After 85 iterations and 160 function evaluations `amoeba` has found values of the coefficients that provide a better fit than the tobit model (154 compared to 157). Notice the coefficients themselves are quite a bit different than the tobit estimates.

Running `simplex` (Barr 1996) from the same starting values we get the following results:

```
. gl fvals = 0
. mat b1 = (.1,5.1)
. simplex ycn X, from(b1) lf(powell) iterate(100)
Iteration 1: Minimum likelihood -447.7554016113281, maximum norm is .14142136
(output omitted)
Iteration 100: Minimum likelihood -164.6005096435547, maximum norm is .07158911
stopped at 100th iteration.
Estimate using simplex, tolerance .00001:
Likelihood value  -164.1095123291016
No. Observations  100
------------------------------------------------------------------------------
     ycn |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
       X |    2.11325          .          .       .              .           .
   _cons |      .3315          .          .       .              .           .
------------------------------------------------------------------------------
. di "Number of function evaluations = " $fvals
Number of function evaluations = 2300
```

After 100 iterations and 2300 evaluations, `simplex` is still at a worse value for the objective function than `amoeba` after 160 function evaluations. The difference may be in the internal setting of parameters of the NM algorithm that control how the simplex evolves.

References

Barr, T. 1996. sg56: An ado-file implementation of a simplex-based maximization algorithm. *Stata Technical Bulletin* 32: 3–27. Reprinted in *Stata Technical Bulletin Reprints*, vol. 6, pp. 134–140.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. 1987. *Numerical Recipes: The Art of Scientific Computing*. New York: Cambridge University Press.

smv3.2 Enhancements to discriminant analysis

Joseph Hilbe, Arizona State University, atjmh@asuvm.inre.asu.edu

The first Stata version of a dichotomous response discriminant analysis program was published in STB-5 (Jan 1992) as smv3. The program was revised with updated code and improved options in STB-34 (Nov 1996) as smv3.1. With suggestions for additional output and corrections from Carlos Ramalheira, Faculty of Medicine, University of Coimbra, Portugal, I am again offering an update, albeit a minor one, to the `discrim.ado` program.

Enhancements include the following additional output with the `predict` option: Percentage positive and negative predicted value, probability of group membership in confusion matrix, Kendall's tau-b, and Cohen's kappa statistic. The output values for false positive and false negative have been corrected.

The above mentioned STB inserts provide examples of program use. The `discrim.hlp` file remains the same as published in STB-34. My thanks go to Dr. Ramalheira for his useful insights and suggested code for the additional statistical output.

snp13	Nonparametric assessment of multimodality for univariate data

Isaas Hazarmabeth Salgado-Ugarte*, Makoto Shimizu, and Toru Taniuchi
University of Tokyo, Faculty of Agriculture, Department of Fisheries, Japan
*Present Address: F.E.S. Zaragoza, U.N.A.M. Biología, México, FAX: (52-5) 773-0151, fes01@tzetzal.dcaa.unam.mx

In previous inserts we presented a series of programs to calculate density estimates for univariate data (Salgado-Ugarte et al. 1993, 1995a). We also introduced a series of practical bandwidth rules for histograms, frequency polygons, and kernel density estimators, including cross-validation techniques (Salgado-Ugarte et al. 1995b). In this insert we present an implementation of the smoothed bootstrap procedure of Silverman (1981) for multimodality assessment.

Silverman test for multimodality

As mentioned in our previous inserts, several procedures to assess the modality of a univariate distribution have been proposed (Good and Gaskins 1980; Hartigan and Hartigan 1985). The test proposed by Silverman uses nonparametric kernel density estimation techniques to determine the most probable number of modes in the underlying density. In what follows we include a brief description of this procedure. The description is based in large part on Izenman and Sommer (1988).

Given a sample $X_1, \ldots, X_n$ from a population having density function f, the expression for a kernel density estimator can be written as

$$\hat{f}(x) = (nh)^{-1} \sum_{j=1}^{n} K((x - X_j)/h), \qquad x \in \Re \tag{1}$$

The choice of the bandwidth h in (1) is an important statistical problem. A very small value of h provides a density estimate that is very noisy (dependent upon the sample values), whereas a very large value for h yields an oversmoothed estimate which removes interesting details. Several rules for the choice of optimal and oversmoothed values of h for several univariate density estimators were presented in Salgado-Ugarte et al. (1995b). It must be emphasized however that the primary concern here is mode counting and not optimal estimation of the smoothing parameter h, although both problems are related (Izenman and Sommer 1988).

In Silverman's test for multimodality, the null hypothesis, H_0^k, states that the true density f possesses at most k modes, whereas the alternative hypothesis, H_1^k, states that f has more than k modes, $k = 1, 2, \ldots$. If we let

$$N(f) = \#\{x : f'(x) = 0, f''(x) < 0\} \tag{2}$$

be the number of modes of f (maxima count), then $H_0^k : N(f) \le k$ and $H_1^k : N(f) > k$. If $\hat{f}_h$ is the kernel density estimator of f with bandwidth h, then a statistic of interest is $N(\hat{f}_h)$, that is, the number of modes in $\hat{f}_h$. Define the kth critical bandwidth as

$$h_{k,crit} = \inf\{h : N(\hat{f}_h) \le k\} \tag{3}$$

that is, the smallest bandwidth that is still consistent with H_0^k. Because Silverman's method strongly depends on the properties of the Gaussian kernel, it is necessary to employ a Gaussian weight function:

$$K(t) = (2\pi)^{-1/2} \exp(-t^2/2), \qquad t \in \Re \tag{4}$$

as the kernel in (1). Under these conditions Silverman found $\hat{f}_h$ is a right-continuous decreasing function of h, and hence that $N(\hat{f}_h) > k$ iff $h < h_{k,crit}$. Therefore, to find $h_{k,crit}$ we count modes in each density estimate $\hat{f}_h$ for different values of h. To find the critical bandwidths Silverman suggests using a simple binary search procedure. If there is an interval where the critical bandwidth is known to fall, then the interval extremes sum divided by two provides a bandwidth for which the number of modes is counted. Izenman and Sommer provide additional guidance in the critical bandwidth finding; when $h = h_{k,crit}$, $\hat{f}_h$, (the estimated density) will display k modes plus a noticeable "shoulder" in its graph, and if h is reduced further, an additional $(k+1)$st mode will appear in place of that shoulder. Such a density with a shoulder employing a critical bandwidth ($\hat{f}_{h_{k,crit}}$) is a critical density. It follows that

$$\Pr_f\{h_{k,crit} > h\} = \Pr\{N(\hat{f}_h) > k | X_1, X_2, \ldots, X_n \text{ is drawn from } f\} \tag{5}$$

Silverman used the ability to sample from a critical density to combine the monotonicity property of $N(\hat{f}_h)$ with the bootstrap to construct a workable test for multimodality.

To assess the significance of a mode count, the following algorithm is used:

1. From the original random sample, $X_1, X_2, \ldots, X_n$, draw n times with replacement to get a bootstrap sample denoted by $X_1^*, \ldots, X_n^*$.
2. Compute a smoothed bootstrap sample, $Y_1^*, Y_2^*, \ldots, Y_n^*$ using

$$Y_j^* = c_k\{X_j^* + h_{k,crit}Z_j\}, \qquad j = 1, 2, \ldots, n, \tag{6}$$

where Z_j is an independent standard Gaussian deviate, and

$$c_k = (1 + [h_{k,crit}/s]^2)^{1/2} \tag{7}$$

is used to rescale the result so that the variance of Y_j^* is equal to the sample variance s^2 of the original data (see Efron, 1982).

3. Use Equations (7), (1), and (4) to form an estimate $\hat{f}^*_{h_{k,crit}}$, of f.
4. Repeat steps 1–3 a large number, B, of times. Let $\hat{f}^{*b}_{h_{k,crit}}$ denote the density estimate for the bth smoothed bootstrap sample.
5. Set

$$I_{k,b} = \begin{cases} 1, & \text{if } N(\hat{f}^{*b}_{h_{k,crit}}) > k \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Then,

$$P_k = B^{-1}\sum_{b=1}^{B} I_{k,b} \tag{9}$$

is the estimated significance level (or p-value) of $h_{k,crit}$.

This testing procedure is repeated for a successively larger number of modes until a sufficiently large p-value is obtained. Silverman offered no suggestions for what “large” might be for this stopping rule. In his article he applied the procedure to the Good and Gaskins (1980) chondrite meteor data (n = 22) and showed that the critical window widths had p-values P_1 =0.08, P_2 =0.05, P_3 =0.79, and P_4 =0.93 and stopped at k = 3 for a trimodal density. In a follow-up paper, Silverman (1983) showed theoretically that the bootstrap test may be conservative. No simulation studies of this test have been published, but this does not limit its value as an exploratory data analytical technique. Izenman and Sommer (1988) suggest that a flexible attitude be taken in applying the test to data with long and complicated tail structures. They commented that there is no reason to expect the sequence of p-values (9) to be monotonically increasing; indeed, Silverman’s own study of the chondrite data illustrates that point. Furthermore, it is possible, depending on the placement of modes, that large fluctuations in the p-values be observed in practice. Based on their experience and on the previous remarks in relation to the conservative nature of the bootstrap test, they suggest applying a flexible stopping rule with a nominal p-value of 0.40 until a detailed study is carried out. They also strongly recommend studying graphical displays of the density estimate near each critical window width and the graphs of the density estimates at the critical windows themselves during the progress of Silverman’s test.

Example

Consider the catfish data we have used in previous inserts. From July, 1980 to August 1981 a total of 2436 individuals of the catfish *Arius melanopus* were collected in the Tampamachoco coastal lagoon in the Northeastern coast of Mexico. Because of the statistical difference that was found among males, females, and juveniles it was considered appropriate to analyze the data by separating the sexes. The organisms of unknown sex (juveniles) were included as a subsample of approximately 50% of the total. In this section we present the analysis of the female and unknown-sex individual’s body length data to assess the multimodality of its distribution. A brief summary of the data is given in Table 1.

Table 1. Number of females and unknown-sex individuals of *Arius melanopus* considered for multimodality assessment.

```
        sex|      Freq.     Percent        Cum.
------------+-----------------------------------
    Females |        556       49.82       49.82
    Unknown |        560       50.18      100.00
------------+-----------------------------------
      Total |       1116      100.00
```

To calculate the kernel density estimates a modified version of the ASH-WARPing procedure described in Salgado-Ugarte et al. (1995a; 1995b) was used. The new program is `warpdenm.ado` (based on `warpdens.ado`) and has the following syntax.

Syntax for warpdenm

`warpdenm` *varname* [`if` *exp*] [`in` *range*] `, bwidth(`*#*`) kercode(`*#*`) mval(`*#*`)` [`step numodes modes npoints gen(`*denvar midvar*`) nograph` *graph options*]

Options

Only the new options are explained here:

`numodes` reports the number of modes (maxima) in the density estimation.

`modes` produces a list of the mode estimates (located at the midpoints used for density calculation).

`npoints` gives the number of points used for estimation.

The `numodes` option in combination with the binary search procedure make it possible to find iteratively the critical bandwidths. To accomplish this task we recommend employing `mval`(30) or another number of shifted histograms producing a convenient number of estimation points, more for larger bandwidths, and fewer for narrow ones. It is possible to use the `npoints` option to see the number of points used for estimation. The precision of the original data scale must be considered as well in order to avoid excessive fractional numbers in the bandwidths.

Use of bandwidth rules

As a first step in the analysis of the modality for this data set we include Table 2 containing results of the binwidth/bandwidth rules introduced in Salgado-Ugarte et al. (1995b). Because as stated before, the multimodality test depends on the use of Gaussian kernel, we will only focus our attention to the results regarding this weight function (last three rows).

Table 2. Smoothing parameter rules for catfish (*Arius melanopus*) length data (females + unknown sex individuals).

```
------------------------------------------------------------
Some practical number of bins and binwidth-bandwidth rules
for univariate density estimation using histograms,
frequency polygons (FP) and kernel estimators
============================================================

Sturges´ number of bins =                         11.1241
Oversmoothed number of bins <=                    13.0687
------------------------------------------------------------
FP oversmoothed number of bins <=                  9.6115
============================================================

Scott´s Gaussian binwidth =                       18.7654
Freedman-Diaconis robust binwidth =               18.3175
Terrell-Scott´s oversmoothed binwidth >=          16.3750
Oversmoothed Homoscedastic binwidth >=            19.9932
Oversmoothed robust binwidth >=                   23.8402
------------------------------------------------------------
FP Gaussian binwidth =                            29.3822
FP oversmoothed binwidth >=                       31.8421
============================================================

Silverman´s Gaussian kernel bandwidth =           12.2995
Haerdle´s ´better´ Gaussian kernel bandwidth =    14.4861
Scott´s Gaussian kernel oversmoothed bandwidth =  15.6340
------------------------------------------------------------
```

Figure 1 displays the Gaussian kernel density estimation by using the optimal bandwidth value from Table 2. The oversmoothed estimate is included in Figure 2. Both figures show a high degree of multimodality and according to the considerations of Terrell and Scott (1985) and Terrell (1990), these modes are strongly suggested as real structures of the data set. They are worthy of additional analysis.

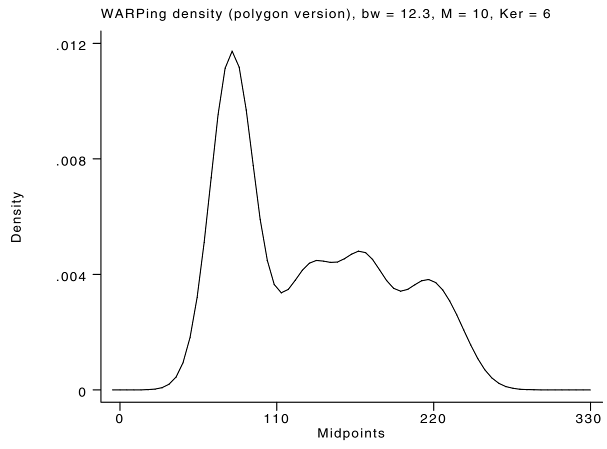

Figure 1. Gaussian kernel density estimate with optimal bandwidth h = 12.3.

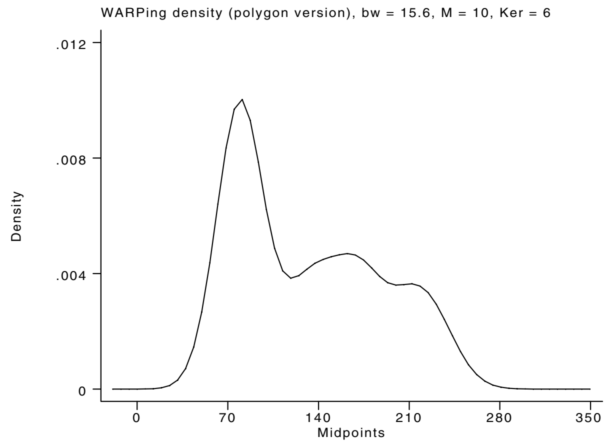

Figure 2. Gaussian kernel density estimate with Scott's oversmoothed bandwidth h = 15.6.

Investigating the optimal bandwidth by means of cross-validation produced the results presented in Tables 3 and 4. The least squares cross-validation (L2CV) function for Gaussian kernel is displayed in Figure 3. Note that this function is almost flat near the minimum. Figure 4 shows the biased cross-validation (BCV) function for the triweight kernel. The minimum is clearer but there is some indication of the existence of an additional local minimum above the oversmoothed bandwidth.

Table 3. Least squares cross-validation score for female-unknown sex catfish data (n = 1116).

```
--------------------------------------------------------------------------------
Least Squares Cross-validation for WARPing density estimation, Gaussian kernel
--------------------------------------------------------------------------------
CV-value = -0.00841106       M-value =     8     Bandwidth =   4.0000
CV-value = -0.00841082       M-value =     7     Bandwidth =   3.5000
CV-value = -0.00840228       M-value =     9     Bandwidth =   4.5000
CV-value = -0.00840097       M-value =     6     Bandwidth =   3.0000
CV-value = -0.00838488       M-value =    10     Bandwidth =   5.0000
```

Table 4. Biased cross-validation score for female-unknown sex catfish data (n = 1116).

```
----------------------------------------------------------------------------
Biased Cross-validation for WARPing density estimation, Triweight kernel
----------------------------------------------------------------------------
Biased Cv-value = 0.00564258   M-value =    10   Bandwidth =  10.0000
Biased Cv-value = 0.00564327   M-value =    11   Bandwidth =  11.0000
Biased Cv-value = 0.00564445   M-value =     9   Bandwidth =   9.0000
Biased Cv-value = 0.00564636   M-value =    12   Bandwidth =  12.0000
Biased Cv-value = 0.00564954   M-value =     8   Bandwidth =   8.0000
```

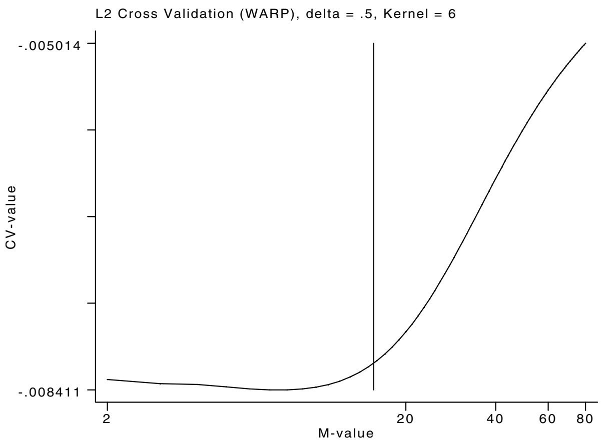

Figure 3. Least squares cross-validation score for female-unknown sex catfish data (n = 1116). The oversmoothed bandwidth is indicated by the line at 15.6.

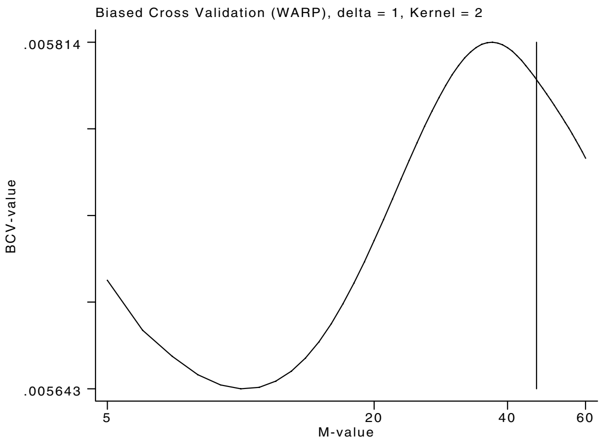

Figure 4. Biased cross-validation score for female-unknown sex catfish data (n = 1116). The rescaled oversmoothed bandwidth is indicated by the line at 46.5.

Note that the cross validation tables and figures were obtained using Pascal programs on a PC using routines described in Salgado-Ogarte (1995b). A `do` file which generates all the graphs in this insert is included on the diskette accompanying STB-38.

The L2CV and BCV (the latter rescaled to Gaussian) optimal bandwidth values were very close (4 and 3.36 respectively). The density estimate with $h = 4$ is presented in Figure 5. The multimodal structure is clearly evident.

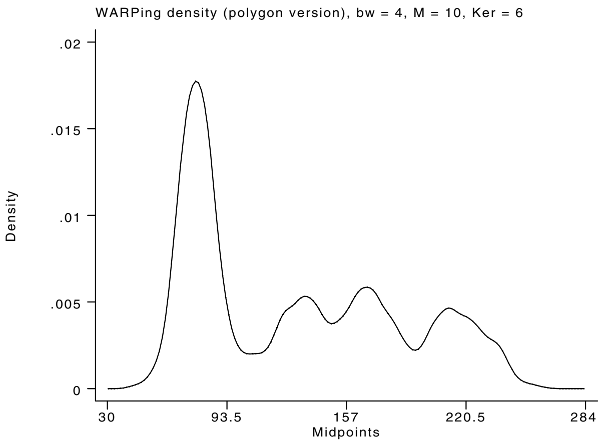

Figure 5. Gaussian kernel density estimate based on the cross-validation optimal bandwidth h = 4 for female-unknown sex catfish data (n = 1116).

Recalling Scott (1992) the agreement of L2CV and BCV should be taken seriously. All the previous results provide a strong evidence for a multimodal distribution of the standard body length of the catfish data set. In what follows we present additional support for this assertion.

Applying the "dip statistic" test for unimodality by using the FORTRAN program (Hartigan and Hartigan 1985) obtained from Statlib and the function provided by Dr. Dario Ringach (personal communication) yielded the value of 0.02782 for the catfish length data. Using an argument similar to that given by Hartigan and Hartigan (1985) for their example, the hypothesis of unimodality was rejected.

Silverman's multimodality test (smoothed bootstrap) calculation

To perform the multimodality test proposed by Silverman it is necessary to generate bootstrapped samples. To do this we wrote a small program used in conjunction with the Stata command `boot`. The program is `bootsam.ado` which performs the calculations required to obtain smoothed bootstrap samples taking into account the equations from steps 2 and 3 of the algorithm described above.

Syntax of boot using bootsam

`boot bootsam, arguments(`*varname critbw*`) iterate(`*#*`)`

where, the option `arguments` allows one to input arguments to the `boot` command: *varname* is the name of the variable containing the original observations from which the sample is to be taken; *critbw* is the value of the critical bandwidth for a given number of modes; and `iterate` refers to the desired number of samples.

Wishing to employ a total of 100 samples for the catfish data and to avoid an excessive number of observations in memory we can simulate two bootstrapped samples of 50 repetitions. For the first 50 we have

```
. set seed 12345
. boot bootsam, arg(blfemin 25.26) iterate(50)
```

The first line sets the seed for the random numbers. A different value for each set of 50 samples is required to have a different series of (pseudo) random numbers. The result of this command is 50 bootstrapped samples from the original variable `blfemin`, using the critical bandwidth for one mode (25.26), according to the expressions of Silverman (1981). The samples contain the variable `ysm`, which is the smoothed bootstrapped sample, the original variable `blfemin` (repeated in each sample), `_rep` which is an indicator variable representing the number of the sample, and `_obs` is the number of observations by sample. If required to save space, variables other than `ysm` and `_rep` can be dropped.

We have automated the last step of Silverman's algorithm in the `silvtest.ado` file. This program calculates the *p*-value of a specified number of modes by estimating the density with a Gaussian kernel for each bootstrapped sample, counting the corresponding modes, and calculating from the total repetitions the fraction of estimates with more modes than the number tested. This command has the following syntax:

Syntax of silvtest

`silvtest` *smvar repndx*`, critbw(`*#*`) mval(`*#*`) nuri(`*#*`) nurf(`*#*`) cnmodes(`*#*`)`
[`nograph` *graph-options*]

Here *smvar* is the smoothed bootstrapped variable and *repndx* is the index of the repetition.

Options

`critbw` is the critical bandwidth for the number of modes to be tested.

`mval` is the number of averaged shifted histograms used to calculate the required density estimations.

`nurf` permits one to specify the final number of replication. It is necessary to input its value to run the program.

`cnmodes` refers to the critical number of modes, that is the number of modes to be tested.

`nuri` permits one to specify the initial number of replication to begin. The default is 1.

`nograph` suppresses the graph.

graph_options are any of the options allowed with `graph, twoway`.

If the user does not provide `critbw`, `mval`, `nurfin`, and `cnmodes`, the program halts and displays an error message on the screen.

This program allows the examination of the kernel density estimate for each bootstrapped sample (Figure 6 displays the estimate for the first sample), counts the correspondent modes and calculates the p-value for the number of modes to be tested. As an example we present here the estimation of the p-value of one mode. In the first place we type the command and options to obtain an output reporting the repetition and its corresponding number of modes. After counting the modes of the last repetition the program displays the p-value including the numbers used for calculation. In that way it is possible to draw another bootstrap sample and to accumulate the number of estimates with more modes that the one tested to calculate the p-value.

```
. silvtest ysm _rep, cr(25.26) m(30) nurf(50) cnm(1)
bs sample     1             Number of modes = 1
bs sample     2             Number of modes = 1
bs sample     3             Number of modes = 1
(output omitted)
bs sample     49            Number of modes = 1
bs sample     50            Number of modes = 1

Critical number of modes =      1

Pvalue =          0 / 50 =    0.0000
```

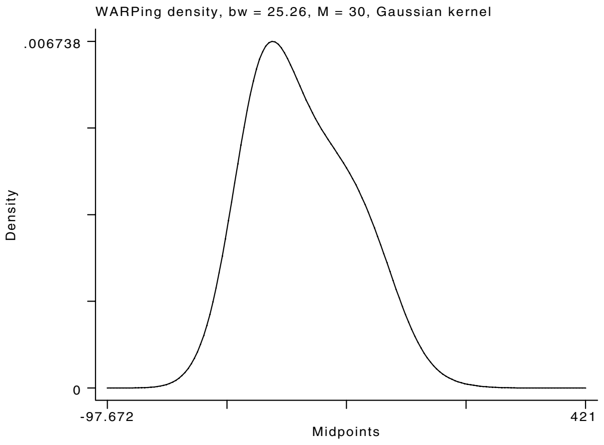

Figure 6. Gaussian kernel density estimate using the critical bandwidth for one mode (25.26) for the first bootstrapped sample.

This procedure is repeated with another bootstrapped 50 samples, and then for other samples using a different critical number of modes. Table 5 shows the critical window widths and their corresponding p-values for the n = 1136 catfish length data carrying out a total of 100 repetitions (B) for each critical bandwidth. These results indicate that the data are consistent with an underlying density having four modes. From this table it can be seen that the sequence of p-values is not strictly monotonically increasing, but after four modes they do not attain a value less than 0.56. This provides solid evidence for the multimodality of this data in addition to the assurance of the number of modes. On the other hand, as mentioned above, this test is known to be rather conservative (Silverman 1983) and although this testing procedure underestimates the number of modes it gives a reliable lower bound for it (Roeder 1990).

Table 5. Critical bandwidths and estimated significance levels for catfish length data (females and unknown sex) $n = 1136$.

Number of modes	Critical bandwidth	p-values
1	25.26	0.00
2	16.56	0.00
3	12.34	0.02
4	3.93	0.89
5	3.19	0.93
6	3.05	0.81
7	2.86	0.90
8	2.79	0.70
9	2.75	0.56

In the catfish data four modes occur for h between 12.34 and 3.93. This range contains the concordant L2CV and BCV optimal h values. Instead of using an intermediate bandwidth (for example h = 8), based on the agreement of the cross-validated rules cited above, we preferred to employ the bandwidth recommended before ($h = 4$). Using the `warpdenm` command it is possible to have a list of the estimated modes by including the `numodes` and `modes` options as follows; with the `npoints` option the number of points used for the estimation are reported.

```
. warpdenm blfemin, b(4) m(10) k(6) numo mo np nog
 Number of modes = 4

---------------------------------------------------------------------------
 Modes in WARPing density estimation, bw = 4, M = 10, Ker = 6
---------------------------------------------------------------------------
 Mode (    1 ) =      76.8000
 Mode (    2 ) =     134.4000
 Mode (    3 ) =     168.0000
 Mode (    4 ) =     211.2000
---------------------------------------------------------------------------
 Number of estimated points = 159
```

The mode at 76.8 corresponds to the individuals of unknown sex, and the three following modes to the adult subpopulations of females. There is some indication of an additional mode above 211.2 but here we will not pursue this possibility.

A multimodality testing procedure related to the Silverman's test is that of Wong (1985). In his report, Wong assesses the effectiveness of his procedure. We hope to present a similar evaluation for the performance of the Silverman test in a future insert.

Some additional comments

We are including the three data sets reported in the literature which have been tested for multimodality with the Silverman's procedure: the chondrite data (`chondri.dta`, from Scott 1992), the Hidalgo's stamp thickness data (`stamp.dta` from Izenman and Sommer 1988), and the galaxies velocities data (`galvel.dta` from Roeder 1990). We apply our programs to all of these data sets and the results are included in Tables 6–9. Putting aside some minor differences in the critical bandwidths (due to differences in the procedure for the calculation of the kernel density estimate) we arrived at the same conclusions as the original reports. We also include the multimodality test for the geyser data (Härdle 1991). Please note that for this data set the number of repetitions for each critical bandwidth is 600, representing the highest value reported for the Silverman test at present and suggest that the duration of the eruptions are bimodal. The reader may try to apply the procedure with our programs on these or any other univariate data set if desired.

Table 6. Critical bandwidths and estimated significance levels for chondrite data (from Scott 1992) $n = 22$.

Number of modes	Critical bandwidth	p-values
1	2.40	0.16
2	1.83	0.06
3	0.69	0.72
4	0.47	0.73

Table 7. Critical bandwidths and estimated significance levels for stamp thickness data (from Izenman and Sommer 1988) $n = 485$.

Number of modes	Critical bandwidth	p-values
1	0.00667	0.00
2	0.00331	0.26
3	0.00300	0.05
4	0.00282	0.00
5	0.00253	0.01
6	0.00246	0.00
7	0.00148	0.52
8	0.00138	0.19
9	0.00105	0.62

Table 8. Critical bandwidths and estimated significance levels for galaxies velocity data (from Roeder 1990) $n = 82$.

Number of modes	Critical bandwidth	p-values
1	3037	0.000
2	2447	0.005
3	920	0.555
4	875	0.203
5	721	0.193
6	664	0.113
7	447	0.343

Table 9. Critical bandwidths and estimated significance levels for geyser data (duration in minutes; from Härdle 1992) $n = 272$.

Number of modes	Critical bandwidth	p-values
1	0.830	0.000
2	0.127	0.495
3	0.084	0.948

The core of the mode counter included as an option in the `warpdenm` program is presented here in the form of an ado-file named `numode`.

Syntax of numode

`numode` *denvar midvar* [`if` *exp*] [`in` *range*] [, `modes`]

where *denvar* is the density or frequency variable and *midvar* contains the corresponding midpoints. The only option of the program is `modes` which permits one to include the list of mode estimates. This small program is useful not only for density estimates but for any frequency-midpoints pairs data.

Acknowledgments

The first author is grateful to the Ministry of Education, Science and Culture of Japan and to the Universidad Nacional Autónoma de México (FES Zaragoza and DGAPA) for their support. The last part of this investigation was supported by the PAPIIT program (project code IN217596) of DGAPA, Universidad Nacional Autónoma de México.

References

Efron, B. 1982. *The Jackknife, the Bootstrap, and Other Resampling Plans*. Philadelphia: Society for Industrial and Applied Mathematics.

Good, I. J. and R. A. Gaskins. 1980. Density estimation and bump-hunting by the penalized likelihood method exemplified by scattering and meteorite data. *Journal of the American Statistical Association* 75: 42–73.

Härdle, W. 1991. *Smoothing Techniques With Implementation in S*. New York: Springer-Verlag.

Hartigan, J. A. and P. M. Hartigan. 1985. The dip test of multimodality. *Annals of Statistics* 13: 70–84.

Izenman, A. J. and C. Sommer. 1988. Philatelic mixtures and multimodal densities. *Journal of the American Statistical Association* 83: 941–953.

Roeder, K. 1990. Density estimation with confidence sets exemplified by superclusters and voids in the galaxies. *Journal of the American Statistical Association* 85: 617–624.

Salgado-Ugarte, I. H., M. Shimizu, and T. Taniuchi. 1993. snp6: Exploring the shape of univariate data using kernel density estimators. *Stata Technical Bulletin* 16: 8–19. Reprinted in *Stata Technical Bulletin Reprints*, vol. 3, pp. 155–173.

—. 1995a. snp6.1: ASH, WARPing, and kernel density estimation for univariate data. *Stata Technical Bulletin* 26: 23–31. Reprinted in *Stata Technical Bulletin Reprints*, vol. 5, pp. 161–172.

—. 1995b. snp6.2: Practical rules for bandwidth selection in univariate density estimation. *Stata Technical Bulletin* 27: 5–19. Reprinted in *Stata Technical Bulletin Reprints*, vol. 5, pp. 172–190.

Scott, D. W. 1992. *Multivariate Density Estimation: Theory, Practice, and Visualization*. New York: John Wiley & Sons.

Silverman, B. W. 1981. Using kernel density estimates to investigate multimodality. *Journal of the Royal Statistical Society, Series B* 43: 97–99.

—. 1983. Some properties of a test for multimodality based on kernel density estimates. In *Probability, Statistics and Analysis*, ed. J. F. C. Kingman and G. E. H. Reuter, 248–249. Cambridge: Cambridge University Press.

Terrel, G. R. 1990. The maximal smoothing principle in density estimation. *Journal of the American Statistical Association* 85: 470–477.

Terrel, G. R. and D. W. Scott. 1985. Oversmoothed nonparametric density estimates. *Journal of the American Statistical Association* 80: 209–214.

Wong, M. A. 1985. A bootstrap testing procedure for investigating the number of subpopulations. *Journal of Statistical Computation and Simulation* 22: 99–112.

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

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	Company:	Smit Consult
Address:	P.O. Box 1169 Nazerath-Ellit 17100, Israel	Address:	Doormanstraat 19 Postbox 220 5150 AE Drunen Netherlands
Phone:	+972 66554254	Phone:	+31 416-378 125
Fax:	+972 66554254	Fax:	+31 416-378 385
Email:	sasconsl@actcom.co.il	Email:	j.a.c.m.smit@smitcon.nl
Countries served:	Israel	Countries served:	Netherlands
Company:	Dittrich & Partner Consulting	Company:	Survey Design & Analysis Services
Address:	Prinzenstrasse 2 D-42697 Solingen Germany	Address:	249 Eramosa Road West Moorooduc VIC 3933 Australia
Phone:	+49 212-3390 99	Phone:	+61 3 59788329
Fax:	+49 212-3390 90	Fax:	+61 3 59788623
Email:	evhall@dpc.de	Email:	rosier@survey-design.com.au
Countries served:	Austria, Germany, Italy	Countries served:	Australia
Company:	Metrika Consulting	Company:	Timberlake Consultants
Address:	Roslagsgatan 15 113 55 Stockholm Sweden	Address:	47 Hartfield Crescent West Wickham Kent BR4 9DW U.K.
Phone:	+46-708-163128	Phone:	+44 181 462 0495
Fax:	+46-8-6122383	Fax:	+44 181 462 0493
Email:	hedstrom@metrika.se	Email:	timberlake@compuserve.com
Countries served:	Baltic States, Denmark, Finland, Iceland, Norway, Sweden	Countries served:	Ireland, U.K.
Company:	Ritme Informatique	Company:	Timberlake Consultants Satellite Office
Address:	34 boulevard Haussmann 75009 Paris France	Address:	Praceta do Comércio, N°13–9° Dto. Quinta Grande 2720 Alfragide Portugal
Phone:	+33 1 42 46 00 42	Phone:	+351 (01) 4719337
Fax:	+33 1 42 46 00 33	Telemóvel:	0931 62 7255
Email:	info@ritme.com	Email:	timberlake.co@mail.telepac.pt
Countries served:	Belgium, France, Luxembourg, Switzerland	Countries served:	Portugal