A publication to promote communication among Stata users

Contents of this issue

- dm48. An enhancement of reshape
- sbe15. Age-specific reference intervals for normally distributed data
- sbe16. Meta-analysis
- sg70. Interquantile and simultaneous-quantile regression
- sg71. Routines to maximize a function
- smv3.2. Enhancements to discriminant analysis
- snp13. Nonparametric assessment of multimodality for univariate data
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 reshape/2 described in this insert seeks to eliminate these limitations, while remaining fully backward
compatible with reshape. In fact, reshape/2 is a fairly extensive rewrite of the code of reshape. Thus, the keyword-based
syntax of reshape/2 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 reshape/2 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 placeholder @. 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  inc_8/0  inc_8/1  inc_8/2
  -------  -----  --------  --------  --------
     1     0     5000     5500     6000
     2     1     2000     2200     3300
     3     0     3000     2000     1000
```

```
   id  year  sex  inc
  -------  -----  -----  -----  -------
     1     80     0     5000
     2     81     1     2000
     2     82     1     3300
     3     81     0     3300
     3     82     0     1000
```

```
   . reshape groups year 80-82
   . reshape vars inc
   . reshape cons id sex
   . reshape long
   (goes from left-form to right)
   . reshape 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 minc/9/0 minc/9/1 fedu finc/9/0 finc/9/1
-------  -----  ----  -----  -------  -------  -----  -------  -------
     1       NY     3     B  30000  32000     H  23000  23700
     2      Phil     1     B  31000  33100     B  34200  35200
     3       SF     2     M  43000  46100     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):

```
   . reshape id hnr
   . reshape cons hcity hnkids
   . reshape groups sex m f, string
   . reshape vars @inc90 @inc91 @edu
   . reshape long
   . sort hnr
   . list, nodisplay noobs
   hnr  hcity  hnkids  inc_90  inc_91  edu  sex
  ------  -------  -----  ------  ------  ----  ----
     1       NY     3  30000  32000     B    m
     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  46100     M    m
```

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

```
   . reshape id hnr sex
   . reshape cons hcity hnkids edu
   . reshape groups year 90-91
   . reshape vars inc@
   . reshape long
   . sort hnr sex year
```
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).

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(s:df 4,s:df 2) centile(2.5 97.5) detail
--- FP Powers ---
<table>
<thead>
<tr>
<th>Cycle</th>
<th>Mean</th>
<th>SD</th>
<th>Deviance</th>
<th>Change</th>
<th>Residual SS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>2</td>
<td>3021.644</td>
<td>0.000</td>
<td>5708.761</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2988.312</td>
<td>-33.332</td>
<td>5094.04</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2988.320</td>
<td>0.008</td>
<td>5094.916</td>
</tr>
</tbody>
</table>
Final deviance = 2988.320 (92 observations).
Power(s) for mean curve = 2.2. Power(s) for SD curve = 1.
```
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 experimenting with real data, it was found that about three cycles are usually sufficient. 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 on $T$. In subsequent cycles the regression is weighted using the inverse square of the estimated SD curve from the previous cycle.

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 experimenting with real data, it was found that about three cycles are usually sufficient.

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 experimenting with real data, it was found that about three cycles are usually sufficient.
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 \texttt{xrigls} command with the \texttt{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 2965.018 -36.625 4969.406
  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 C/2/./5 C/9/7/./5/, any : rename @_gl @_cv
  gen SD_cv=S_sd gawks, c(11) s(11) so xlabel 11(Standard deviation)
  > xlabel gap(5)
```

Figure 2. 95% RI and median where SD is an estimated constant.
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(111111)
> s(iiiiiii) sort xlabel ylabel(/2/0/,/4/0/,/6/0/,/8/0/,/1/0/0)
> l(/"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 \( \text{RI} \) 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 l0=0 g=-2*se0
. for C2.5 C50 C97.5, any : gen u0=0 g2=2*se0
. graph C2.5_gls 1C2.5 wC2.5 C50_gls 1C50 wC50 C97.5_gls 1C97.5 wC97.5 gawks
> if gawks>27, c(11111111) s(iiiiiiii) xlabel ylabel
> l(/"Bi-parietal diameter (mm)="/) gap(5)
```
Technical note

The variables $M_{\text{gls}}$ and $S_{\text{gls}}$ created by \texttt{xrigls} are the estimated mean and SD curves. When the data are approximately normally distributed, use of \texttt{xriml} with the option \texttt{dist/(n/)} gives very similar results to those obtained from \texttt{xrigls}. The variables $M_{\text{ml}}$ and $S_{\text{ml}}$, created by \texttt{xriml}, are also the estimated mean and SD curves. However, when the exponential-normal or modulus-exponential-normal distributions are selected in \texttt{xriml} using the options \texttt{dist/(en/)} or \texttt{dist/(men/)} respectively, $M_{\text{ml}}$ and $S_{\text{ml}}$ are then the estimated median and scale parameter curves (Wright and Royston 1997b).

Syntax of \texttt{xrigls}

\begin{verbatim}
xrigls yvar xvar [if] [in] range [ , major_options minor_options ]
\end{verbatim}

The \texttt{major_options} (most used options) are

- \texttt{alpha(#)} specifies the significance level for testing between degrees of FP for the mean and SD curves. Default is 0.05.
- \texttt{centile(# [# [# ...]])} defines the centiles of $yvar$ | $xvar$ required. Default is 3 and 97 (i.e. a 94% reference interval).
- \texttt{cv} specifies the $s$ curve to be modeled as a coefficient of variation.
- \texttt{detail} displays the final regression models for the mean and SD curves.
- \texttt{fp(m:term, s:term)} specifies fractional polynomial models in $xvar$ for the mean and SD curves. \texttt{term} is of the form \texttt{powers} # | \texttt{df} #. The phrase \texttt{powers} is optional. The powers should be separated by spaces, for example \texttt{fp(m: powers 0 1, s: powers 2)}. If \texttt{powers} or \texttt{df} are not given for any curve, the default is 4 \texttt{df} for the mean and 2 \texttt{df} for the SD. \texttt{#} specifies that the degrees of freedom for the best-fitting FP model are to be at most \texttt{#} for the curve in question. The best-fitting powers are then determined from the data.

The \texttt{minor_options} (less used options), in alphabetic order, are

- \texttt{covars(m:mcovars, s:scovars)} includes \texttt{mcovars} (\texttt{scovars}) variables as predictors in the regression model for the mean (SD) curves.
- \texttt{cycles(#)} determines the number of fitting cycles. The default value of \texttt{#} 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.

Figure 6. Confidence limits for centiles of biparietal diameter when gestational age is greater than 27 weeks.
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 \( p < \alpha \).

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

tcs (m: mopts, s: sopts) determines the regression options for the mean and SD regression models. For example,

tcsp (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


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 $\theta$ 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 $\chi^2_{k-1}$ 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 $\tau^2$ is a moment estimator put forward by DerSimonian and Laird (1986):

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

An overall random-effects estimate can then be calculated as

$$\hat{\theta}_R = \frac{\sum_{i=1}^{k} w_i^* \hat{\theta}_i}{\sum_{i=1}^{k} w_i^*}$$

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 $\hat{\tau}^2$ 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:

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

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 $\text{ebest}_i$ is

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

Syntax

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

```stata
meta theta setheta [if exp] [in range] [ weight | eform print ebayes level(#) graph (f|x|e) 
    id(strvar) fmult(#) boxysca(#) boxshad(#) oline 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 \( \chi^2 \) 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 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
```
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

| 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

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

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.
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(0.5,1.5) xline(1)
> boxsh(2) b("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**

<table>
<thead>
<tr>
<th>Study</th>
<th>Fixed Est</th>
<th>Rand Est</th>
<th>Lower</th>
<th>Upper</th>
<th>z_value</th>
<th>p_value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weseley</td>
<td>0.672</td>
<td>0.564</td>
<td>0.564</td>
<td>0.800</td>
<td>-4.455</td>
<td>0.000</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Study</th>
<th>Fixed Est</th>
<th>Rand Est</th>
<th>Lower</th>
<th>Upper</th>
<th>z_value</th>
<th>p_value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.50</td>
<td>0.400</td>
<td>0.889</td>
<td>-2.337</td>
<td>0.011</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 0.63 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 0.85
Landesma 66.34 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 0.36 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  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


---

**sg70**  | Interquantile and simultaneous-quantile regression

---

William Gould, StataCorp, wgould@stata.com

Linear regression measures \( E(y|\mathbf{x}) = \mathbf{x}\beta \). Quantile regression focuses on the quantiles instead of the expected value and measures \( Q_{\alpha}(y|\mathbf{x}) = \mathbf{x}\beta_{\alpha} \). For instance, \( Q_{.50}(y|\mathbf{x}) = \mathbf{x}\beta_{.50} \) 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 \( Q_{.25}(y|\mathbf{x}) = \mathbf{x}\beta_{.25} \) reflecting the lower quartile of the data or \( Q_{.75}(y|\mathbf{x}) = \mathbf{x}\beta_{.75} \) reflecting the upper quartile. If the distribution of \( y|\mathbf{x} \) has constant variance, then the \( Q_{.25} \) and the \( Q_{.75} \) relationship will simply parallel the \( Q_{.50} \) relationship in that all estimated coefficients except for the intercepts will be roughly the same. If the coefficients differ, this is evidence of heteroskedasticity.

Heteroskedasticity—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

\[
Q_{.25}(y|\mathbf{x}) = 100 - 10 \cdot A - 20 \cdot B \\
Q_{.50}(y|\mathbf{x}) = 160 - 10 \cdot A - 10 \cdot B \\
Q_{.75}(y|\mathbf{x}) = 220 - 10 \cdot A - 4 \cdot B
\]
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] [in range] [, quantiles(#) reps(#) nolog level(#) ]
```

```
sqreg depvar [varlist] [if] [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)`, and `quantiles(.25 .50 .75)` specifies three equations; a .50 quantile (median) regression is to be added.

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

This default is arguably too small. \texttt{reps(100)} would perform 100 bootstrap replications. \texttt{reps(1000)} would perform 1,000.

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

\texttt{level(\#)} specifies the confidence level in percent for the confidence interval of the coefficients.

\textbf{Remarks}

If you are not familiar with quantile regression, please see \cite{qreg}.

Consider a quantile-regression model where the \( q \)th 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

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

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

The other new command, \texttt{sqreg}, is like \texttt{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 \texttt{qreg}. \texttt{sqreg} differs from \texttt{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

\begin{verbatim}
. qreg y x1 x2, q(.25)
. qreg y x1 x2, q(.75)
\end{verbatim}

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

\begin{verbatim}
. iqreg y x1 x2, q(.25 .75)
\end{verbatim}

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:

\begin{verbatim}
. sqreg y x1 x2, q(.25 .75)
. test [q25]x1 = [q75]x1
\end{verbatim}

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

Finally, \texttt{sqreg} can estimate quantiles singly,

\begin{verbatim}
. sqreg y x1 x2, q(.5)
\end{verbatim}
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:

```stata
. regress ln_wage ed tenure

Source | SS df MS Number of obs = 2377
---------+--------------------------------------------------
Model | 182.39998 2 91.099993 F( 2, 2374) = 321.14
Residual | 673.42838 2374 .283667071 R-squared = 0.2129
---------+--------------------------------------------------
Total | 855.62136 2376 .360101040 Root MSE = .5326

ln_wage | Coef. Std. Err. t P>|t| [95% Conf. Interval]
---------+--------------------------------------------------
 ed | 0.093355 .004465 20.091 0.000 .084916 .097636
 tenure | -0.265811 .005992 4.455 0.000 .023497 .008086
 _cons | .5400002 .0599684 9.229 0.000 .423138 .6568664
---------+--------------------------------------------------

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 dlny/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:

```stata
. 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 = 906.36535
(output omitted)
```

```
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 | 0.0947895 .0045872 20.664 0.000 .0857942 .1037849
 tenure | -.0308667 .0050605 14.944 0.000 .0265461 .0362773
 _cons | .4268735 .061496 6.941 0.000 .3052821 .5484649
---------+--------------------------------------------------

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:

```stata
. qreg ln_wage ed tenure, q(.25) nolog
```

```
25 Quantile Regression
Number of obs = 2377
Raw sum of deviations 848.9267 (about 1.4766921)
Min sum of deviations 712.3449 Pseudo R2 = 0.1609

ln_wage | Coef. Std. Err. t P>|t| [95% Conf. Interval]
---------+--------------------------------------------------
 ed | 0.0850433 .0048776 17.436 0.000 .0757486 .0943388
 tenure | .0337707 .0024232 13.937 0.000 .0289189 .0386224
 _cons | .2415063 .065603 3.681 0.000 .1128612 .3701514
---------+--------------------------------------------------
```
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:

<table>
<thead>
<tr>
<th>Variable</th>
<th>25th percentile</th>
<th>50th percentile</th>
<th>75th percentile</th>
</tr>
</thead>
<tbody>
<tr>
<td>ed</td>
<td>0.85</td>
<td>0.94</td>
<td>0.99</td>
</tr>
<tr>
<td>tenure</td>
<td>0.34</td>
<td>0.30</td>
<td>0.24</td>
</tr>
<tr>
<td>_cons</td>
<td>0.241</td>
<td>0.427</td>
<td>0.693</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>ln_wage</th>
<th>Coef. Std. Err. t  P&gt;</th>
<th>t</th>
<th>[95% Conf. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ed</td>
<td>0.090176</td>
<td>0.005267</td>
<td>18.798</td>
</tr>
<tr>
<td>tenure</td>
<td>0.0234354</td>
<td>0.0020309</td>
<td>10.567</td>
</tr>
<tr>
<td>_cons</td>
<td>0.6931128</td>
<td>0.0690686</td>
<td>9.972</td>
</tr>
</tbody>
</table>
```

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 - [q50]ed = 0.0

| ln_wage | Coef.Std. Err. t P>|t|[95% Conf. Interval] |
|---------|------------------|---------------|---|-----------------|
| (1)     | 0.139742 .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 [q50]tenure = [q75]tenure, notest 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 Intequantile Regression Number of obs = 2377
   bootstrap(100) SEs .75 Pseudo R2 = 0.1496
   .25 Pseudo R2 = 0.1609

| ln_wage | Coef. Std. Err. t P>|t|[95% Conf. Interval] |
|---------|------------------|---------------|---|-----------------|
| ed      | 0.139742 .005438 2.570 0.010 .0033105 .026438 |
| tenure  | -.0094252 .0024666 -3.821 0.000 -.0142622 -.0045833 |
| _cons   | .4516065 .0712686 6.337 0.000 .3185514 .5846616 |
```

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 $b_{75,ed} - b_{25,ed}$ 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 \( \epsilon \) 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
```

and, correspondingly, would have produced the median regression

```
iqreg y x1 x2
```

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

```
iqreg y x1 x2
```

```
(bootstrap ....................)
```

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
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

<table>
<thead>
<tr>
<th>coefficient</th>
<th>true value</th>
<th>average value</th>
<th>average width</th>
<th>95% coverage</th>
<th>CI for coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>reps(20)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b1</td>
<td>0</td>
<td>0.0042</td>
<td>0.6928</td>
<td>94.4</td>
<td>92.8–95.7</td>
</tr>
<tr>
<td>b2</td>
<td>0</td>
<td>-0.0059</td>
<td>0.6972</td>
<td>94.9</td>
<td>93.3–96.2</td>
</tr>
<tr>
<td>intercept</td>
<td>1.3490</td>
<td>1.3498</td>
<td>0.5212</td>
<td>93.4</td>
<td>91.8–94.9</td>
</tr>
<tr>
<td>reps(300)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b1</td>
<td>0</td>
<td>-0.0032</td>
<td>0.7011</td>
<td>96.8</td>
<td>95.5–97.8</td>
</tr>
<tr>
<td>b2</td>
<td>0</td>
<td>0.0063</td>
<td>0.7010</td>
<td>95.9</td>
<td>94.5–97.0</td>
</tr>
<tr>
<td>intercept</td>
<td>1.3490</td>
<td>1.3453</td>
<td>0.5266</td>
<td>96.0</td>
<td>94.6–97.1</td>
</tr>
</tbody>
</table>

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 \( b_2 = 0 \):

<table>
<thead>
<tr>
<th>coefficient</th>
<th>true value</th>
<th>average value</th>
<th>average width</th>
<th>95% coverage</th>
<th>CI for coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>reps(20)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b1</td>
<td>1.3490</td>
<td>1.3516</td>
<td>0.7486</td>
<td>93.4</td>
<td>91.7–94.9</td>
</tr>
<tr>
<td>b2</td>
<td>0</td>
<td>-0.0057</td>
<td>0.7063</td>
<td>95.0</td>
<td>93.5–96.3</td>
</tr>
<tr>
<td>intercept</td>
<td>0.8094</td>
<td>0.8112</td>
<td>0.4746</td>
<td>94.3</td>
<td>92.7–95.7</td>
</tr>
<tr>
<td>reps(300)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b1</td>
<td>1.3490</td>
<td>1.3428</td>
<td>0.7557</td>
<td>97.7</td>
<td>96.6–98.5</td>
</tr>
<tr>
<td>b2</td>
<td>0</td>
<td>0.0087</td>
<td>0.7099</td>
<td>95.4</td>
<td>93.9–96.6</td>
</tr>
<tr>
<td>intercept</td>
<td>0.8094</td>
<td>0.8058</td>
<td>0.4764</td>
<td>95.3</td>
<td>93.8–96.7</td>
</tr>
</tbody>
</table>

Performance

Estimating bootstrapped standard errors is never quick but those who have used \texttt{bsqreg} will be pleasantly surprised. In fact, even when your interest is not in comparing quantiles, you will want to use \texttt{sqreg} as an alternative to \texttt{bsqreg}. \texttt{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 \texttt{sqreg} and \texttt{iqreg}, it is. The older \texttt{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 \texttt{bsqreg}.

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

<table>
<thead>
<tr>
<th>Replications</th>
<th>60 MHz Pentium Windows 95</th>
<th>120 MHz Pentium Unix</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>bsqreg</td>
<td>sqreg</td>
</tr>
<tr>
<td>20</td>
<td>2.55</td>
<td>2.16</td>
</tr>
<tr>
<td>50</td>
<td>6.01</td>
<td>5.00</td>
</tr>
<tr>
<td>100</td>
<td>11.89</td>
<td>9.80</td>
</tr>
<tr>
<td>250</td>
<td>30.09</td>
<td>24.00</td>
</tr>
<tr>
<td>500</td>
<td>62.11</td>
<td>47.66</td>
</tr>
<tr>
<td>1000</td>
<td>132.62</td>
<td>95.00</td>
</tr>
</tbody>
</table>
All times are reported in seconds. The commands executed were

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

with the automobile data loaded into memory.

References


---

<table>
<thead>
<tr>
<th>sg71</th>
<th>Routines to maximize a function</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Christopher Ferrall, Queen’s University, Kingston, Ontario, Canada, FAX 613-545-6668, <a href="mailto:ferrallc@post.queensu.ca">ferrallc@post.queensu.ca</a></td>
</tr>
</tbody>
</table>

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

```stata
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.

*iimax* 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, \(\beta\) is a \((k \times 1)\) vector of unknown parameters, and \(\epsilon\) is the \((N \times 1)\) vector of disturbance terms. The ordinary least squares (OLS) estimate \(\hat{\beta}\) of \(\beta\) solves:

\[
\hat{\beta} = \underset{\beta}{\arg\min} \left( y - X\beta \right)^2
\]

Of course, the solution is \(\hat{\beta} = \left( X'X \right)^{-1} X'y\). 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.

```stata
. set obs 100
. gen X = (_n-50)/10
. gen y = -2.0 + 3.0 * X + 5*invnorm(uniform())
. regress y X
```

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>Number of obs = 100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>7699.47768</td>
<td>1</td>
<td>7699.47768</td>
<td>Prob &gt; F = 0.0000</td>
</tr>
<tr>
<td>Residual</td>
<td>2068.67513</td>
<td>98</td>
<td>21.1089299</td>
<td>R-squared = 0.7882</td>
</tr>
<tr>
<td>Total</td>
<td>9768.15281</td>
<td>99</td>
<td>98.6662102</td>
<td>Adj R-squared = 0.7861</td>
</tr>
</tbody>
</table>

| y     | Coef. | Std. Err. | t     | P>|t|     | [95% Conf. Interval] |
|-------|-------|-----------|-------|--------|---------------------|
| X     | 3.039786 | .1591642 | 19.098 | 0.000  | 2.723933 3.355642 |
| _cons | -2.34626 | .4595135 | -5.106 | 0.000  | -3.25817 -1.434391 |

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

```stata
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**:

```stata
. global nlist = "beta:X beta:_cons"
. global yvar = "y"
. matrix b0 = (.1 .5 .1)
. matrix colnames b0 = `nlist`
```
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 converges at the default criterion with SSE reduced to 2068.8961. 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
```

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 \(\text{ycn} - \max(X/3,0)\). First, consider the tobit estimates:

```
. tobit ycn X, ll(0)
```

```
Tobit Estimates
Number of obs = 100
chi2(1) = 107.27
Prob > chi2 = 0.0000
Pseudo R2 = 0.2806
Log Likelihood = -137.38223

ycn | Coef. Std. Err. t P>|t| [95% Conf. Interval]
-------------+--------------------------------------------------
   X | 3.154331  .580773  5.609  0.000   2.443282   3.864384
   _cons | -2.6174  1.048694 -2.496  0.014  -4.698217  -0.536839
-------------+--------------------------------------------------
   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`):

```stata
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
gen double `yh' = sum(`tob')
scalar `2' = `yh'[M]
```

Starting from the same vector `b0` used with `myols`, and the initial 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.

```stata
. global yvar = "ycn"
. matrix sig = (3.0)
. matrix btob = `b0' , `sig'
. qui amoeba tobin btob yout b1 . . E-5
. quasi tobin b1 yout bfinal g h . . bhhh
```

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:

```stata
qui replace `e' = abs(`yvar' - max(`e',0))
qui egen `sd' = sum(`e')
scalar `2' = -`sd'
/* 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:

```stata
. 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:
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:

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


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.
that is very noisy (dependent upon the sample values), whereas a very large value for $h$
mode counting and not optimal estimation of the smoothing parameter
estimators were presented in Salgado-Ugarte et al. (1995b). It must be emphasized however that the primary concern here is
estimators, including cross-validation techniques (Salgado-Ugarte et al. 1995b). In this insert we present an implementation of
1993, 1995a). We also introduced a series of practical bandwidth rules for histograms, frequency polygons, and kernel density
as
of the Gaussian kernel, it is necessary to employ a Gaussian weight function:

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

where $x$ is a real number.

To find the critical bandwidths Silverman suggests using a simple binary search procedure. If there is an interval where the

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

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:

$$K(t) = (2\pi)^{-1/2} e^{-t^2/2}, \quad t \in \mathbb{R}$$

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 ($f_{h_{k,crit}}$) is a critical density. It follows that

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

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

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

where $Z_j$ is an independent standard Gaussian deviate, and

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

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 $\hat{f}_{h_{k, \text{crit}}}$, of $f$.

4. Repeat steps 1–3 a large number, $B$, of times. Let $\hat{f}_{h_{k, \text{crit}}}^b$ denote the density estimate for the $b$th smoothed bootstrap sample.

5. Set

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

Then,

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

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

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>
<thead>
<tr>
<th>sex</th>
<th>Freq</th>
<th>Percent</th>
<th>Cum.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Females</td>
<td>596</td>
<td>49.82</td>
<td>49.82</td>
</tr>
<tr>
<td>Unknown</td>
<td>560</td>
<td>49.18</td>
<td>100.00</td>
</tr>
<tr>
<td>Total</td>
<td>1156</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>

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 `warpden.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).**

<table>
<thead>
<tr>
<th>Smoothing Parameter Rules</th>
<th>Sturges' Number of Bins</th>
<th>FP Oversmoothed Number of Bins</th>
<th>Scott's Gaussian Binwidth</th>
<th>Freedman-Diaconis Robust Binwidth</th>
<th>Terrell-Scott's Oversmoothed Binwidth</th>
<th>Oversmoothed Homoscedastic Binwidth</th>
<th>Oversmoothed Robust Binwidth</th>
<th>FP Gaussian Binwidth</th>
<th>FP Oversmoothed Binwidth</th>
</tr>
</thead>
</table>

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.
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 \((n = 1116)\).

<table>
<thead>
<tr>
<th>Least Squares Cross-validation for WARPing density estimation, Gaussian kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV-value = -0.00641106</td>
</tr>
<tr>
<td>CV-value = -0.00641082</td>
</tr>
<tr>
<td>CV-value = -0.00640228</td>
</tr>
<tr>
<td>CV-value = -0.00640097</td>
</tr>
<tr>
<td>CV-value = -0.00638488</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Biased Cross-validation for WARPing density estimation, Triweight kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biased CV-value = 0.00564258</td>
</tr>
<tr>
<td>Biased CV-value = 0.00564327</td>
</tr>
<tr>
<td>Biased CV-value = 0.00564445</td>
</tr>
<tr>
<td>Biased CV-value = 0.00564636</td>
</tr>
<tr>
<td>Biased CV-value = 0.00564954</td>
</tr>
</tbody>
</table>
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(#) nurf(#) nuri(#) cmodes(#)
    [ 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.
- `cmodes` 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 `cmodes`, 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) n(30) muf(50) cmn(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
```

![Gaussian kernel density estimate using the critical bandwidth for one mode (25.26) for the first bootstrapped sample.](image)

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) $n = 1136$.

<table>
<thead>
<tr>
<th>Number of modes</th>
<th>Critical bandwidth</th>
<th>$p$-values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25.26</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>16.56</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>12.34</td>
<td>0.02</td>
</tr>
<tr>
<td>4</td>
<td>3.93</td>
<td>0.89</td>
</tr>
<tr>
<td>5</td>
<td>3.19</td>
<td>0.93</td>
</tr>
<tr>
<td>6</td>
<td>3.05</td>
<td>0.81</td>
</tr>
<tr>
<td>7</td>
<td>2.86</td>
<td>0.90</td>
</tr>
<tr>
<td>8</td>
<td>2.79</td>
<td>0.70</td>
</tr>
<tr>
<td>9</td>
<td>2.75</td>
<td>0.56</td>
</tr>
</tbody>
</table>

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 \texttt{warpdem} command it is possible to have a list of the estimated modes by including the \texttt{numodes} and \texttt{modes} options as follows; with the \texttt{npoints} option the number of points used for the estimation are reported.

\[ \texttt{warpdem blfemin, b(4) m(10) k(6) numo mo np nog} \]

\texttt{Number of modes = 4}

\begin{verbatim}
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
\end{verbatim}

\texttt{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.

\textbf{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 (\texttt{chondri.dta}, from Scott 1992), the Hidalgo’s stamp thickness data (\texttt{stamp.dta} from Izenman and Sommer 1988), and the galaxies velocities data (\texttt{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 (H"ardle 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.

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

<table>
<thead>
<tr>
<th>Number of modes</th>
<th>Critical bandwidth</th>
<th>(p)-values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.40</td>
<td>0.16</td>
</tr>
<tr>
<td>2</td>
<td>1.83</td>
<td>0.06</td>
</tr>
<tr>
<td>3</td>
<td>0.69</td>
<td>0.72</td>
</tr>
<tr>
<td>4</td>
<td>0.47</td>
<td>0.73</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Number of modes</th>
<th>Critical bandwidth</th>
<th>(p)-values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00667</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>0.00331</td>
<td>0.26</td>
</tr>
<tr>
<td>3</td>
<td>0.00300</td>
<td>0.05</td>
</tr>
<tr>
<td>4</td>
<td>0.00282</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>0.00253</td>
<td>0.01</td>
</tr>
<tr>
<td>6</td>
<td>0.00246</td>
<td>0.00</td>
</tr>
<tr>
<td>7</td>
<td>0.00148</td>
<td>0.52</td>
</tr>
<tr>
<td>8</td>
<td>0.00138</td>
<td>0.19</td>
</tr>
<tr>
<td>9</td>
<td>0.00105</td>
<td>0.62</td>
</tr>
</tbody>
</table>
Table 8. Critical bandwidths and estimated significance levels for galaxies velocity data (from Roeder 1990) \( n = 82 \).

<table>
<thead>
<tr>
<th>Number of modes</th>
<th>Critical bandwidth</th>
<th>( p )-values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3037</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>2447</td>
<td>0.005</td>
</tr>
<tr>
<td>3</td>
<td>920</td>
<td>0.555</td>
</tr>
<tr>
<td>4</td>
<td>875</td>
<td>0.203</td>
</tr>
<tr>
<td>5</td>
<td>721</td>
<td>0.193</td>
</tr>
<tr>
<td>6</td>
<td>664</td>
<td>0.113</td>
</tr>
<tr>
<td>7</td>
<td>447</td>
<td>0.343</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Number of modes</th>
<th>Critical bandwidth</th>
<th>( p )-values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.830</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>0.127</td>
<td>0.495</td>
</tr>
<tr>
<td>3</td>
<td>0.084</td>
<td>0.948</td>
</tr>
</tbody>
</table>

The core of the mode counter included as an option in the \texttt{warpedm} program is presented here in the form of an ado-file named \texttt{numode}.

**Syntax of numode**

\[ \texttt{numode} \texttt{denvar} \texttt{midvar} \ [\texttt{if} \ \texttt{exp}] \ [\texttt{in} \ \texttt{range}] \ [\texttt{,} \ \texttt{modes}] \]

where \textit{denvar} is the density or frequency variable and \textit{midvar} contains the corresponding midpoints. The only option of the program is \texttt{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**


STB categories and insert codes

Inserts in the STB are presently categorized as follows:

**General Categories:**
- an announcements
- cc communications & letters
- dm data management
- dt datasets
- gr graphics
- in instruction
- ip instruction on programming
- os operating system, hardware, & interprogram communication
- qs questions and suggestions
- tt teaching
- zz not elsewhere classified

**Statistical Categories:**
- sbe biostatistics & epidemiology
- sed exploratory data analysis
- sg general statistics
- smv multivariate analysis
- snp nonparametric methods
- sqc quality control
- sqv analysis of qualitative variables
- sr robust methods & statistical diagnostics
- srd robust methods & statistical diagnostics
- sss social science & psychometrics
- std time-series, econometrics
- sxi survey sampling
- sxe experimental design
- ssw survival analysis
- ssi simulation & random numbers
- ssp survey sampling
- szz not elsewhere classified

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.

<table>
<thead>
<tr>
<th>Company</th>
<th>Address</th>
<th>Phone</th>
<th>Fax</th>
<th>Email</th>
<th>Countries served</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applied Statistics &amp; Systems</td>
<td>P.O. Box 1169, Nazerath-Ellit 17100, Israel</td>
<td>+972 66554254</td>
<td>+972 66554254</td>
<td><a href="mailto:sasconsl@actcom.co.il">sasconsl@actcom.co.il</a></td>
<td>Israel</td>
</tr>
<tr>
<td>Smit Consult</td>
<td>Doormanstraat 19, Netherlands</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dittrich &amp; Partner Consulting</td>
<td>Prinzenstrasse 2, Solingen, Germany</td>
<td>+49 212-3390 99</td>
<td>+49 212-3390 90</td>
<td><a href="mailto:evhall@dpc.de">evhall@dpc.de</a></td>
<td>Austria, Germany, Italy</td>
</tr>
<tr>
<td>Survey Design &amp; Analysis</td>
<td>249 Eramosa Road West, Moorooduc VIC 3933, Australia</td>
<td>+61 3 59788329</td>
<td>+61 3 59788623</td>
<td><a href="mailto:rosier@survey-design.com.au">rosier@survey-design.com.au</a></td>
<td></td>
</tr>
<tr>
<td>Metrika Consulting</td>
<td>Roslagsgatan 15, Stockholm, Sweden</td>
<td>+46-708-163128</td>
<td>+46-8-6122383</td>
<td><a href="mailto:hedstrom@metrika.se">hedstrom@metrika.se</a></td>
<td>Baltic States, Denmark, Finland, Iceland, Norway, Sweden</td>
</tr>
<tr>
<td>Timberlake Consultants</td>
<td>47 Hartfield Crescent, West Wickham, Kent BR4 9DW U.K.</td>
<td>+44 181 462 0495</td>
<td>+44 181 462 0493</td>
<td><a href="mailto:timberlake@compuserve.com">timberlake@compuserve.com</a></td>
<td></td>
</tr>
<tr>
<td>Ritme Informatique</td>
<td>34 boulevard Haussmann, Paris, France</td>
<td>33 1 42 46 00 42</td>
<td>33 1 42 46 00 33</td>
<td><a href="mailto:info@ritme.com">info@ritme.com</a></td>
<td>Belgium, France, Luxembourg, Switzerland</td>
</tr>
<tr>
<td>Timberlake Consultants</td>
<td>Praceta do Comércio, N°13–9ºDto. Quinta Grande, Alfragide Portugal</td>
<td>+351 (01) 4719337</td>
<td>+931 62 7255</td>
<td><a href="mailto:timberlake.co@mail.telepac.pt">timberlake.co@mail.telepac.pt</a></td>
<td>Portugal</td>
</tr>
</tbody>
</table>