STATA
TECHNICAL
BULLETIN

September 1995
STB-27

A publication to promote communication among Stata users

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

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

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

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

Contents of this issue

page

an55.	New Stata for Macintosh released	2
an56.	Stata for Windows 95 and Stata for WindowsNT released	2
ip9.	Repeat Stata command by variable(s)	3
snp6.2.	Practical rules for bandwidth selection in univariate density estimation	5
ssa7.	Analysis of follow-up studies	19
ssa8.	Analysis of case−control and prevalence studies	26

an55 New Stata for Macintosh released

William Gould and Chinh Nguyen, Stata Corporation, FAX 409-696-4601

A new version of Stata 4.0 for Macintosh, called Stata 4.0+ for Macintosh, is now shipping. This new version is important in two ways:

1. The new Stata for Macintosh provides native Power Mac support.
2. On both the Power Mac and 680x0 series computers, the new Stata for Macintosh provides the new Stata windowed interface first seen in Stata for Windows, including the spreadsheet editor.

Compared to the existing Stata for Macintosh product, the interface to Stata 4.0+ has been completely redesigned. In addition, all aspects of dealing with the operating system—saving and printing graphs, saving and printing logs, scrolling, and so on—have been completely rewritten. That is, we rebuilt Stata for Macintosh from the ground up.

In addition to the new features, our early timings indicate that the 4.0+ version is faster:

Computer	Test	Old Stata	New Stata 4.0+	Ratio
Power Mac 6100	Test 1	33.56	2.47	13.6
	Test 2	2940.00	78.22	37.6
	Sort	92.10	5.43	17.0
	Poisson	5.13	.17	30.2
68040 Quadra 630	Test 1	6.78	8.54	.79
	Test 2	402.47	355.53	1.13
	Sort	17.85	16.50	1.08
	Poisson	1.50	.47	3.19

Test 1: `set obs 30000`, `set seed 1001`, make 10 uniformly distributed random variables.

Test 2: Test 1 followed by `gen byte y=uniform()>.5`, and then `quietly logit y x*` ten times.

Sort: `set obs 50000`, `set seed 1001`, `gen u=uniform()`, `sort u`.

Poisson: `use auto` followed by `quietly poisson rep78 mpg weight`.

Timings were obtained from `rmsg`. Also note, in the old Stata 4.0 for Macintosh these reported timings were incorrect. The timings were calculated as if they were based on a 100-tick per second clock when Macintoshes actually use a 60-tick/second clock. To obtain the correct timings with the prior Stata for Mac, reported timings were multiplied by 100/60.

The timings for the Power Mac compare 4.0+ for Power Mac to the old Stata.noFPU running in emulation mode. Obviously, most of the improvement is due to elimination of emulation mode but, as the 68040 timings show, Stata 4.0+ is in general faster, too. On the 68040, Stata 4.0+ performed more slowly in Test 1. Exploration of that result revealed that the `uniform()` function is slower in Stata 4.0+. `uniform()` is based on bit manipulation and this is the only reduced performance we have found.

In the tests, all `.ado` files were preloaded before execution so that the disk I/O times did not affect the execution time, although I/O is markedly faster in 4.0+. The old Stata for Mac not only loaded, but ran ado-files very slowly. The code responsible for this has been replaced. Stata 4.0+ runs ado-files faster. The performance improvement for the Poisson test is due to the more rapid rate at which Stata 4.0+ executes ado-file code.

It is also worth noting that the old Stata for Macintosh did not poll for the break key or yield processing time to other tasks often enough. For instance, in the case of `sort`, it never polled! This has been fixed. Stata 4.0+'s times are not only better but the program itself is more responsive to breaks and yielding processing time to other applications.

We strongly recommend obtaining this upgrade. For Stata 4.0 users, the new Stata 4.0+ for Macintosh diskettes are included when you purchase the new *Getting Started with Stata for Macintosh* manual ($30).

an56 Stata for Windows 95 and Stata for WindowsNT released

William Gould and Alan Riley, Stata Corporation, FAX 409-696-4601

Two new products, Stata for Windows^{tm} 95 and Stata for WindowsNT^{tm} are now shipping. While Stata for Windows 3.1 will run under Windows 95, we expect Windows 95 users will want to switch to a Windows 95 native version of Stata. Here is a summary of our current Stata for Windows offerings:

- **Stata for Windows 3.1**.
  Runs under: Windows 3.1, Windows 95.
  Intended user: Windows 3.1.
  Description: This is the product we began shipping in January 1995 and that we continue to ship. It is a 32-bit, Windows 3.1 application.

- **Stata for Windows 95**.
  Runs under: Windows 95, WindowsNT.
  Intended user: Windows 95 or single user of WindowsNT.
  Description: This is a Windows 95 native, 32-bit application and, as such, provides preemptive multitasking and support for long filenames. It also has a Windows 95 look and feel.
- **Stata for WindowsNT**.
  Runs under: WindowsNT and Windows 95 clients of WindowsNT.
  Intended user: Multiuser and/or networked WindowsNT and WindowsNT/Server serving WindowsNT and Windows 95 clients.
  Description: This product is, in effect, an extended version of Stata for Windows 95 for dealing with the multiuser and network aspects of WindowsNT. Stata for WindowsNT is intended for multiuser, networked sites and will support both WindowsNT and Windows 95 clients.

Existing users of Stata for Windows 3.1 can obtain Stata for Windows 95 for $30, including shipping within the U.S., from us.

Stata for WindowsNT is a new product and, since it is explicitly a multiuser Stata for Windows, has the same pricing as Stata for Unix.

Stata for Windows 95 runs 10% to 15% faster than does Stata for Windows 3.1 under Windows 3.1. Whereas Stata itself consumes no more memory, Windows 95 does. We estimate the additional memory consumption to be somewhere between 1 and 2 megabytes. Our experiments indicate that on an 8 megabyte computer, you can allocate about 5.6 megabytes (`/k5600`) without inducing paging. Our early experiences also indicate that Windows 95 multitasks very well. On a 16-megabyte Pentium, we ran six simultaneous Stata sessions, each with a 1-megabyte data area and each running certification do-files. All six ran at reasonable speed and interactive use of other Windows 95 applications was instant.

ip9 Repeat Stata command by variable(s)

Patrick Royston, Royal Postgraduate Medical School, London, FAX (011)-44-181-740-3119

The `by` *varlist* feature in Stata is powerful, but works with only a small number of Stata commands (the help files make clear which). Here I present a new command, `byvar`, which extends `by` to almost any Stata command and adds new facilities.

For example, the `swilk` command allows you to use the Shapiro–Wilk W statistic to test variable(s) for departure from a normal (Gaussian) distribution. Suppose you had a grouping variable called `agegroup`, and you wanted Shapiro–Wilk tests of `height` for each value of `agegroup`. `swilk` does not support `by`, so the natural syntax of

```
. by agegroup : swilk height
```

does not work. (In fact, it appears to work in that it does not "crash", but it does not give you the results you want.)

Using the new `byvar` command, you could enter

```
. byvar agegroup : swilk height
```

A major limitation of `by` is that there is no easy way to store the results of each execution of the Stata command after each of the levels of the by-variable(s) have been processed. `byvar` remedies this and gives you two ways to store such results: in global macros and in new variables. `byvar` allows you to capture results from global macros (such as `$S_1`), the `_result()` function, regression coefficients and their standard errors. In addition it can tabulate such results in a convenient manner.

The syntax of `byvar` is

`byvar` *varlist* [, *options*] : *Stata_cmd*

where the *options* are

coef(*coeflist*) generate macro(*mlist*) missing pause result(*rlist*) se(*selist*) store tabulate

First I present some examples of uses of `byvar`. Details of the *options* and some remarks are given later.

Examples

A data file `igg.dta` was included on the STB-21 disk to accompany *sg26*, an insert on fractional polynomials (Royston and Altman 1994). This file contains data relating serum immunoglobulin IgG concentrations in children to their age. Here I use the data to illustrate the use of `byvar` to test for non-normality of IgG in each of three equal-sized age groups, and to regress IgG on age for each age group.

```
. use igg
. sort age
. generate int agegroup=group(3)
. byvar agegroup : swilk igg
-> agegroup==1
                   Shapiro-Wilk W test for normal data
 Variable |    Obs          W        V         z   Pr > z
 ---------+-----------------------------------------------
      igg |    100     0.97525     2.043      1.585  0.05645
-> agegroup==2
                   Shapiro-Wilk W test for normal data
 Variable |    Obs          W        V         z   Pr > z
 ---------+-----------------------------------------------
      igg |     99     0.93344     5.450      3.759  0.00009
-> agegroup==3
                   Shapiro-Wilk W test for normal data
 Variable |    Obs          W        V         z   Pr > z
 ---------+-----------------------------------------------
      igg |     99     0.97882     1.734      1.221  0.11108
. byvar agegroup, tabulate : swilk igg
nothing to generate, store or tabulate
r(198);
```

The last command failed because, in order to tabulate output, we must first define the quantities that are to be tabulated (or stored). `swilk` leaves behind several numbers in `$S_#` macros, including the number of observations in `$S_1`, the value of the Shapiro–Wilk statistic in `$S_2` and its *p*-value in `$S_5`.

This is how we tabulate the results with appropriate column labels and save the results in new variables.

```
. byvar agegroup, macro(S_1=#obs S_2=W-statistic S_5=P-value) tabulate generate : swilk igg
 agegroup |          #obs   W-statistic       P-value
----------+-----------------------------------------
        1 |           100     .97525101     .05645398
        2 |            99      .9334361     .00008516
        3 |            99     .97881738     .11108145
. describe
Contains data from igg.dta
  Obs:   298 (max=  1283)
 Vars:     7 (max=    99)
Width:    26 (max=   200)
  1. igg          float  %9.0g               IgG (g/l)
  2. age          float  %9.0g               Age (years)
  3. y            float  %9.0g               Square root of IgG
  4. agegroup     int    %8.0g
  5. _M_1         float  %9.0g               #obs by agegroup
  6. _M_2         float  %9.0g               W-statistic by agegroup
  7. _M_3         float  %9.0g               P-value by agegroup
Sorted by:
Note:  Data has changed since last save
```

We now regress `igg` on `age` and display the regression coefficients and their standard errors. (This example is for illustration only!)

```
. byvar agegroup, result(9=mean_sq_error) coef(age) se(age) tabulate : regress igg age
 agegroup | mean_sq_error       _b[age]      _se[age]
----------+-----------------------------------------
        1 |     1.6354381     2.1290027     .48805156
        2 |     1.6933643     .59993447     .28241839
        3 |     2.4147575     1.1737551     .40803034
```

Options

macro(*mlist*) stores the values of global macros which are named in *mlist*. The macros must evaluate to numbers (strings are not allowed). The macro names must be separated by space(s). You may append a label, preceded by an = sign, to each macro name; this will be used to label the corresponding column of output (if the tabulate option is used) or variable (if the generate option is used). The label may be no longer than thirteen characters and must not contain spaces, commas, colons or equals signs. Example: macro(S_1=number_of_obs S_2=Shapiro--Wilk).

result(*rlist*) stores the values of _result() whose arguments are given in *rlist*. Individual items may be labelled as with the macro() option.

coef(*clist*) stores the regression coefficients for variables named in *clist*. Individual items may be labelled as with the macro() option.

se(*slist*) stores the standard errors of regression coefficients for variables named in *slist*. Individual items may be labelled as with the macro() option.

generate creates new variable(s) corresponding to the quantities named in the macro(), result(), coef() and se() options. The new variables are called _M_#, _R_#, _C_# and _S_#, respectively. Sequence numbers (#) correspond to the items stored. For example, macro(S_1 S_3) generate would create variables called _M_1 and _M_2 containing the values of macros $S_1 and $S_3, respectively, which are “left behind” by each execution of *Stata_cmd*. Results are stored according to the combinations of values of the variables in *varlist*.

store stores results corresponding to the quantities named in the macro(), result(), coef() and se() options in global macros whose names begin with M#_, R#_, C#_, S#_, respectively. The #’s are sequence numbers which correspond to the numbers of items stored. These suffixes are followed by integer codes which index the combinations of values of the variables in *varlist*. For example, macro(S_1 S_3) would create macros called $M1_1, $M1_2, ... containing successive values of macro $S_1. Similarly, $M2_1, $M2_2, ... would contain successive values of macro $S_3.

tabulate displays the results in tabular form, suppressing the output (if any) from *Stata_cmd*.

missing causes *Stata_cmd* to be executed even when a combination of values of any of the variables in *varlist* involves a missing value. The idea is the same as for the missing option in Stata’s tabulate command.

pause pauses output after each execution of *Stata_cmd*.

Remarks

In programming byvar, I have attempted to solve an awkward problem: how to incorporate an if phrase, if one is specified in *Stata_cmd*, when filtering *Stata_cmd* according to values in *varlist*. I have done so by searching for if in the part of *Stata_cmd* which precedes the first comma if one is present, or in the whole of *Stata_cmd* if not. There may be types of *Stata_cmd* for which this will not work correctly, but so far none have been encountered.

References

Royston, P. and D. G. Altman. 1994. sg26: Using fractional polynomials to model curved regression relationships. *Stata Technical Bulletin* 21: 11–23.

snp6.2 | Practical rules for bandwidth selection in univariate density estimation

Isaías Hazarmabeth Salgado-Ugarte, Makoto Shimizu, and Toru Taniuchi,
University of Tokyo, Faculty of Agriculture, Department of Fisheries, Japan
FAX (011)-81-3-3812-0529, EMAIL fes01@tzetzal.dcaa.unam.mx

The choice of bandwidth (smoothing parameter) is one of the central problems of density estimation. As we noted in previous inserts (Salgado-Ugarte et al. 1993, 1995), there are several ways to select an appropriate value for this parameter for histograms, frequency polygons (FPs), averaged shifted histograms (ASH/WARP estimators) and kernel estimators. Some of these selection methods focus on the optimal number of intervals, while others approximate the optimal bin width by minimizing an error measurement under specified conditions.

In this insert, we survey a variety of methods for selecting the bandwidth for univariate density estimation. We also present several programs that determine useful reference values for the bandwidth when analyzing densities by means of histograms, FPs, and kernel density estimators, including the average shifted histogram (ASH) and the more general weighted averages of rounded points (WARP). In addition, we include a new, integrated version of our previous programs for univariate density estimation.

Histogram rules for number of bins and bin width choice

Probably the most famous rule for determining the number of intervals for histogram density estimation was proposed by Sturges (1926). The rule is based on the ability to divide a normally distributed variable into classes so the expected class frequencies comprise a binomial series for any sample size, n, that is a power of two (Doane 1976). Technically, Sturges's rule is a procedure to choose the number of intervals, although Sturges explicitly refers to the choice of a class interval. According to Sturges's suggestion, the number of bins, k, is determined by

$$k = 1 + \log_2 n$$

Sturges's formulation is widely recommended in introductory statistics texts. It has become a guideline for researchers, and it is often used as a default in statistical programs even when it is inappropriate. For instance, this rule is not applicable when the data arise from a nonsymmetric, multimodal, or otherwise non-Gaussian distribution (Doane 1976, Scott 1992). Sturges's formula can be adjusted for skewness by adding bins. The number of additional bins is approximated by $\log_2(1 + \widehat{\gamma}\sqrt{n/6})$, where $\widehat{\gamma}$ is an estimate of the standardized skewness coefficient (Doane 1976). For exploratory work, Emerson and Hoaglin (1983) note that this adjustment involves calculations that could be troublesome without a computer. A more serious drawback is the nonresistance of the skewness coefficient.

Histogram bin width rules

Scott (1979) derived a formula to calculate the asymptotically optimal bin width, where the criterion of optimality is the minimum integrated squared error (MISE) of the histogram. Scott's formula requires prior knowledge of the true density function, a rare event in real data analysis. Therefore, adopting the Gaussian density as a reference, he proposed the formula

$$\widehat{h} = 3.5\widehat{\sigma}n^{-1/3}$$

where $\widehat{h}$ is the estimated bin width and $\widehat{\sigma}$ is an estimate of the standard deviation of the data.

Scott also analyzed the performance of this rule when it is applied to three reference non-Gaussian distributions: a skewed distribution (log normal), a heavy-tailed distribution (Student's t), and a bimodal distribution (mixture of two normals). From his simulations, Scott concluded that the Gaussian reference rule

1. oversmooths a log normal density. However, for skewness indexes less than or equal to one, the difference between the estimated and true optimal bandwidths is less than 30 percent.
2. is insensitive to moderate kurtosis.
3. oversmooths bimodal data when the distance between the modes is greater than two. With distinctly bimodal data, Scott's rule is not adequate.

More recently, Scott (1992) has provided correction factors for $\widehat{h}$, accounting for skewness and kurtosis.

A more robust rule has been proposed by Freedman and Diaconis (1981a,b). This rule replaces the estimated standard deviation in Scott's rule with a multiple of the interquartile range (IQR). The Freedman–Diaconis (F–D) rule is

$$\widehat{h} = 2(\text{IQR})n^{-1/3}$$

Several authors have compared the performance of the Sturges, Scott, and F–D rules (Emerson and Hoaglin 1983, Scott 1992; see also the technical note in the Stata Reference Manual, 1995) leading to the following consensus:

1. The Scott and Sturges rules closely agree for samples between 50 and 500.
2. For larger samples, Sturges's rule gives too few bins, leading to oversmoothing.
3. In general, non-Gaussian densities require more bins.

4. The F–D rule calls for narrower intervals (35 percent more bins) than does Scott's rule.
5. From an exploratory point of view, the most interesting feature of the Scott and F–D rules is their dependence on $n^{-1/3}$. In other words, the number of intervals is a function of $n^{1/3}$, a transformation that lies between $\log n$ and $\sqrt{n}$ on the ladder of powers.

Oversmoothed rules

The rules described above provide a simple and useful starting point, but they are not the ultimate answer to the question. Recent research has focused on finding data-based procedures that minimize the MISE or related quantities like the asymptotic mean integrated squared error (AMISE). The procedures described below are some of the fruits of this research. (A more detailed review is provided by Scott, 1992).

Terrell and Scott (1985) showed that, conditional on some data-based knowledge of the scale of the unknown density, there exists a useful upper bound for the width of histogram bins. There is no theoretical lower bound on h as the unknown density can be arbitrarily rough. Terrell's and Scott's formula for the upper bound is

$$h_{OS} = \frac{x_{\max} - x_{\min}}{(2n)^{1/3}} \geq h_O$$

where $x_{\max} - x_{\min} \equiv R$ is the sample range, h_O is the optimal bandwidth, and h_{OS} is the *oversmoothed bandwidth*, that is, the upper bound for h. This formula can be re-expressed as a rule for the number of bins:

$$k_{OS} = \frac{R}{h_O} \geq \frac{R}{h_{OS}} = (2n)^{1/3}$$

Choosing a bin width greater than or equal to h_{OS}, or, equivalently, using no more than k_{OS} bins will produce an oversmoothed estimate. Terrell and Scott conclude that the oversmoothing rules give nearly optimal results for a variety of smooth densities and produce good density estimates.

Terrell (1990) refined these rules further in his development of the maximal smoothing principle. When the variance of the underlying distribution is constant, Terrell's formula for the homoscedastic oversmoothing bandwidth is

$$h_{OS} = 3.729\sigma n^{-1/3} \geq h_0$$

Terrell also derives a robust homoscedastic oversmoothing bandwidth:

$$h_{OS} = 2.603(\text{IQR})n^{-1/3} \geq h_0$$

This latter formula is especially useful with skewed data.

The conservative, oversmoothed density estimates generated by these rules are less likely to display spurious structure. When structural features appear in these conservative estimates, the analyst can have a high degree of confidence that the apparent structure is authentic. Of course, these procedures may fail to detect structures that can only be found using more specialized tests (Terrell 1990).

Frequency polygon rules for the number of bins and for bin width choice

In spite of early criticism (Fisher 1932, 1958) of frequency polygons—that is, the representation of the density as the linear interpolation of the midpoints of a histogram with uniform bin width—the work of Scott (1985b) on the theoretical properties of univariate and bivariate frequency polygons (FP) has demonstrated that they produce much better estimates than the histogram (Scott 1992). The improved results provided by the FP are an important aid in finding the minimum AMISE and, thus, in estimating the optimal number of bins and bin width.

Compared to the histogram, the FP

1. is a better approximation to continuous densities with linear interpolation over wider bins;
2. loses efficiency when the underlying density is discontinuous;
3. is more sensitive to errors in bandwidth choice, particularly when $h > h_0$. On the other hand, quite a large error in bin width for the FP is required before its MISE is worse than that of the best histogram MISE.

These differences are reflected in the resulting Gaussian reference rule for the FP,

$$\widehat{h} = 2.15\widehat{\sigma} n^{-1/5}$$

The estimate of the standard deviation may be a robust one, such as IQR/1.349 (or the F-pseudosigma). This rule also can be adjusted by taking into account modified skewness and kurtosis factors (Scott 1992).

As in the case of histograms, it is possible to define lower bounds for the bin width or upper bounds for the number of bins. The FP rule for the oversmoothed number of bins is

$$\frac{x_{\max} - x_{\min}}{h_0} \geq \left(\frac{147}{2}n\right)^{1/5}$$

A different version of the oversmoothed problem leads to the corresponding bin width rule

$$h_{OS} \equiv 2.33\sigma n^{-1/5} \geq h_O$$

The small difference between the oversmoothed rule for FPs and the FP Gaussian rule suggests that the FP-oversmoothed rule may be used instead of a Gaussian rule when it is difficult to explicitly solve the variational problem (Scott 1992).

Rules for kernel bandwidth choice

In his monograph on density estimation, Silverman (1986) discusses several rules for choosing the bandwidth, h, when using kernel density estimators. One approach is the test graph method (Silverman 1978), which consists of drawing the second derivative of the density estimate, $\widehat{f}$, for various values of h and choosing the bandwidth corresponding to the graph with "rapid well defined fluctuations not fully hiding the systematic variation". Although some subjectiveness is involved, it appears that the test graphs amplify the variation in the density estimates, thus the choice of an appropriate bandwidth is not very difficult in practice. Nevertheless, Silverman recognizes that, because of its dependence on subjective judgments, the test graph method is useful mainly as a check on the results from other methods.

In addition to the test graph method, Silverman proposed using a standard distribution as a reference, in a manner similar to Scott's (1979) use of a reference distribution for the histogram. For instance, if a Gaussian kernel is employed, the optimal bandwidth is estimated by

$$\widehat{h} = 1.06\widehat{\sigma}n^{-1/5}$$

Silverman analyzed the performance of this rule when confronted with non-Gaussian distributions and arrived at conclusions similar to Scott's: this rule

1. oversmooths heavily skewed data;
2. shows little sensitivity to kurtosis (using the lognormal and t distributions); and
3. oversmooths more as the distribution becomes more strongly bimodal.

Silverman also suggested replacing the standard deviation in this rule with the interquartile range, as follows:

$$\widehat{h} = .79(\text{IQR})n^{-1/5}$$

This formula performs better in skewed and long-tailed distributions, but increases oversmoothing in the bimodal case. As a third alternative, Silverman proposed the adaptive rule:

$$\widehat{h} = 1.06An^{-1/5}$$

where

$$A = \min(\widehat{\sigma}, \text{IQR}/1.349)$$

Härdle prefers this adaptive optimum rule and calls it the "better rule of thumb" (Härdle 1991).

It may be worth noting that the IQR, which is calculated by Stata, is slightly different from the *fourth-spread* (Tukey 1977, Hoaglin 1983, Frigge et al. 1989), and many authors use the fourth-spread as their preferred robust measure of spread. In practice, the difference between the IQR and the fourth-spread decreases as the sample size increases (Hamilton 1992).

Silverman suggests an additional adjustment, reducing the factor 1.06 to 0.9 in the formula above, that is,

$$\widehat{h} = .9An^{-1/5}$$

In Silverman's simulations with a Gaussian kernel, this rule provided an MISE within 10 percent of the optimum for the long-tailed, asymmetric, and bimodal distributions he considered (Silverman 1986).

Oversmoothed rule for kernels

Based on previous research (Scott and Terrell 1987, Terrell 1990) and using the variance as the measure of scale, Scott (1992) derived the following oversmoothing rule for kernel density estimators:

$$h_{os} = 3\left[\frac{R(K)}{35\sigma_K^4}\right]^{1/5} \sigma n^{-1/5}$$

where $R(K)$ is the "roughness" of the kernel and σ_K^4 is the squared kernel variance. These measures are constant characteristics of each kernel. Table 1 lists the roughness and variance values for some common kernels.

With this table, it is possible to calculate the oversmoothing rule for the listed kernels. For instance, the rule for the biweight kernel is

$$h_{os} = 3\sigma n^{-1/5}$$

while the rule for the Gaussian kernel is

$$h_{os} = 1.144\sigma n^{-1/5}$$

This latter rule produces bins that are 8 percent wider than those determined from the Gaussian reference rule, using the factor 1.06.

Table 1. Kernel roughnesses and variances for common kernels
(adapted from Scott 1992)

Kernel	$R(K)$	σ_K^2
uniform	1/2	1/3
triangle	2/3	1/6
Epanechnikov	3/5	1/5
biweight	5/7	1/7
triweight	350/429	1/9
Gaussian	$0.5/\sqrt{\pi}$	1
cosinus	σ^2/16	1 - 8/π^2

(Note: the kernels are supported on [-1,1] except for the Gaussian kernel, according to the equations of Härdle 1991 and Scott 1992.)

Least squares cross-validation

Cross-validation (CV) is a well-known procedure for automatically choosing the smoothing parameter. While maximum likelihood can be used to calculate the CV estimate of the smoothing parameter, it is more common to use least-squares CV (L2CV). The least-squares approach was suggested by Rudemo (1982) and Bowman (1984) and is based on a very simple idea. Consider the integrated squared error (ISE) as a measure of the distance, d_I, between the estimated density, $\widehat{f}_h$, and the true density. This distance can be written as a function of the smoothing parameter, h,

$$d_I(h) = \int (\widehat{f}_h - f)^2(x)dx = \int \widehat{f}_h^2(x)dx - 2\int (\widehat{f}_h f)(x)dx + \int f^2(x)dx$$

Note that the first term of this expression can be calculated from the data and the last term does not depend on either the estimated density or h. Thus only the cross-product term in the middle of the expression must be estimated.

The principle of least-squares CV is to minimize the first and second terms of this distance measure with respect to h. L2CV uses the formula for the expectation of an additional and independent observation X,

$$\int (\widehat{f}_h f)(x)dx = E_X[\,\widehat{f}_h(X)]$$

Since an additional data set generally is not available, the *leave one out estimate* is defined as

$$E_X[\,\widehat{\widehat{f}_h}(X)] = n^{-1}\sum_{i=1}^{n}\widehat{f}_{h,i}(X_i)$$

Combining expressions, the L2CV estimate can be written as

$$L2CV(h) = \int \widehat{f}_h^2(x)dx - \frac{2}{n}\sum_{i=1}^{n}\widehat{f}_{h,i}(X_i)$$

Scott and Terrell (1987) showed that the L2CV is an unbiased cross-validation criterion. Härdle (1991) provided an algorithm for computing L2CV, however this algorithm is quadratic in n, the number of observations, a drawback that motivated the search for a more efficient calculation method. In this regard, Silverman (1986) proposed the use of a fast Fourier transform algorithm. Scott and Terrell (1987) used a modified ASH procedure. Härdle (1991) presented an efficient algorithm based on the WARP generalization of ASH. This algorithm is linear in n.

Biased cross-validation

Taking a different approach, Scott and Terrell (1987) suggested choosing h to minimize the asymptotic mean integrated squared error (AMISE). They found this estimator to be biased using the L_2-norm, thus they named it the biased cross-validation (BCV) estimator. Härdle (1991) presented the derivation of the general expression for the BCV along with a full set of computational expressions and an algorithm for calculating the BCV estimator.

Scott and Terrell (1987) compared the performance of the unbiased L2CV and BCV using simulated data. They found that

1. For small samples (n=25), approximately half of the estimated BCV functions had no local minima, although for $n > 40$ all the estimates had a local minimum;
2. BCV had a smaller standard deviation than L2CV;
3. If the underlying density was asymmetric or had heavy tails (Cauchy, lognormal, or Gaussian mixtures), BCV tended to oversmooth. L2CV produced better estimates despite its greater average dispersion.

These results give some guidance in the choice of an estimator. If the true density is asymmetric, then the L2CV estimator should be chosen. Otherwise, the BCV estimator is preferred.

Scott (1992) concluded that, the BCV and L2CV procedures are powerful tools for choosing the bin width for histograms and FP's and the bandwidth for kernel estimators. Scott recommended the use of a $\log(h)$ plot to reveal possible failures of the procedures (no local minimum for BCV or a degenerate $h = 0$ for L2CV). If the two procedures produce markedly different estimates of the bandwidth, Scott suggested choosing the estimate that produces less local noise, especially near the peaks.

Implementation in Stata

As the previous sections have shown, implementing bandwidth selection in Stata is straightforward with the exception of the cross-validated bandwidth estimators. Setting these latter approaches to the side for the moment, we have written `bandw` to display the more easily calculated estimates. The syntax of `bandw` is

`bandw` *varname* [`if` *exp*] [`in` *range*]

`bandw` displays a table with the results of the following rules (the number of estimates displayed appears in parentheses):

1. histogram number of bins rules (2),
2. FP oversmoothed number of bins rule (1),

3. histogram bin width rules (5),
4. FP bin width rules (2), and
5. kernel bandwidth rules (3),

To illustrate `bandw`, we use two data sets introduced in our previous insert, *snp6.1*: the snowfall data (Parzen 1979, Härdle 1991, Scott 1992) and the catfish data consisting of standard body length measures (Salgado-Ugarte 1985). (See *snp6.1* for more details on these data sets.)

```
. use bufsnow, clear

. bandw snow

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

Sturges´ number of bins =                           6.9773
Oversmoothed number of bins <=                      5.0133
-------------------------------------------------------------
FP oversmoothed number of bins <=                   5.4091
=============================================================

Scott´s Gaussian binwidth =                        20.8641
Freedman-Diaconis robust binwidth =                17.4413
Terrell-Scott´s oversmoothed binwidth >=           20.2262
Oversmoothed Homoscedastic binwidth >=             22.2292
Oversmoothed robust binwidth >=                    22.6999
-------------------------------------------------------------
FP Gaussian binwidth =                             22.2680
FP oversmoothed binwidth >=                        24.1323
=============================================================

Silverman´s Gaussian kernel bandwidth =             9.3215
Haerdle´s ´better´ Gaussian kernel bandwidth =     10.9787
Scott´s Gaussian kernel oversmoothed bandwidth =   11.8487
-------------------------------------------------------------
```

Note that the entries in this table are separated according to the five types of rules.

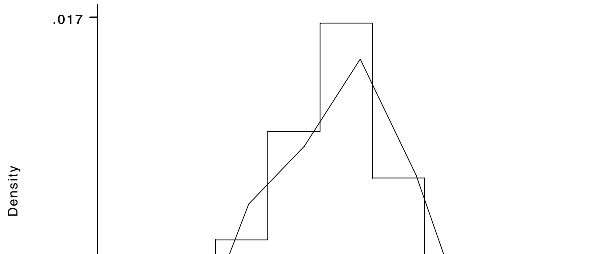

Figure 1. Gaussian reference rule estimates, snowfall data

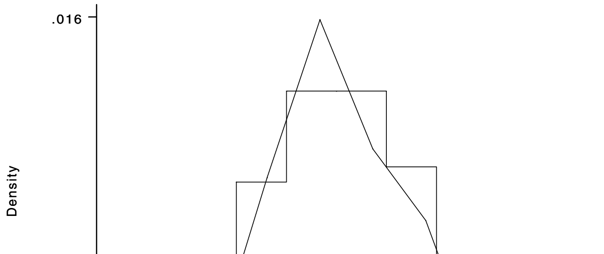

Figure 2. Oversmoothed estimates, snowfall data

Figure 1 displays a histogram and frequency polygon for the snowfall data using the Gaussian reference rule to select the bin width. Both the histogram and the FP were calculated using the revised programs for WARPing density estimation discussed below. ASH/WARP density estimation overcomes problems associated with the choice of origin of the histogram. As a consequence, our programs do not permit the user to override the default choice of origin. Figure 1 gives little indication of the multimodality of these data, although both estimates suggest the possibility of an additional "bump" to the left of the mode.

Figure 2 displays a histogram and FP based on the oversmoothed bin widths. We took a conservative approach and used the largest oversmoothed bin width. As expected, this approach produces very smooth density estimates. However, in the oversmoothed FP, there is a hint of a bump to the right of the mode. Thus, it seems worthwhile to employ additional methods (for example that of Silverman 1981, 1983) to search for a more complex structure.

The estimated bandwidths for the catfish data are listed below:

```
. use catfish, clear

. bandw bodlen

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

Sturges´ number of bins =                      12.2521
Oversmoothed number of bins <=                 16.9595
----------------------------------------------------------
FP oversmoothed number of bins <=              11.2383
==========================================================

Scott´s Gaussian binwidth =                    15.0376
Freedman-Diaconis robust binwidth =            16.0466
Terrell-Scott´s oversmoothed binwidth >=       13.2079
Oversmoothed Homoscedastic binwidth >=         16.0215
Oversmoothed robust binwidth >=                20.8847
----------------------------------------------------------
FP Gaussian binwidth =                         26.1322
FP oversmoothed binwidth >=                    28.3200
==========================================================

Silverman´s Gaussian kernel bandwidth =        10.9391
Haerdle´s ´better´ Gaussian kernel bandwidth =   12.8838
Scott´s Gaussian kernel oversmoothed bandwidth =  13.9048
----------------------------------------------------------
```

Figures 3 and 4 display histograms and FPs for the catfish data using the Gaussian reference rule and the oversmoothed rule, respectively. Both sets of estimates indicate the data have a complex multimodal structure. At least three modes are easily identified, and the oversmoothed results provide strong evidence that these modes are authentic and not artifacts. Further analysis would naturally focus on characterizing these features of the data distribution. See Izenman and Sommers (1990) for an example of the strategy to follow. Other recent accounts of multimodality assessment include Comparini and Gori (1986), Roeder (1990), and Müller and Sawitzki (1991).

As Terrell (1990) suggested, you can use the oversmoothed rules to produce conservative estimates of both histograms and frequency polygons. Alternatively, you can use kernel estimators to explore the bandwidths suggested by `bandw`. With the factors listed in Table 1, we can calculate the oversmoothed bandwidth for non-Gaussian weight functions. A simpler approach is to convert the Gaussian oversmoothed bandwidth to the corresponding bandwidth for any of the kernels listed in Table 2 of our previous insert on ASH-WARPing estimators (*snp6.1*).

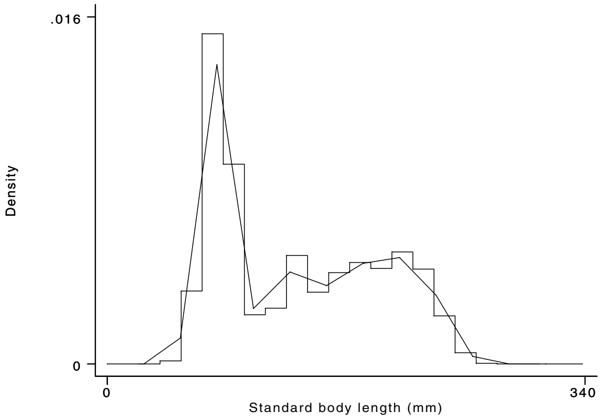

Figure 3. Gaussian reference rule estimates, catfish data

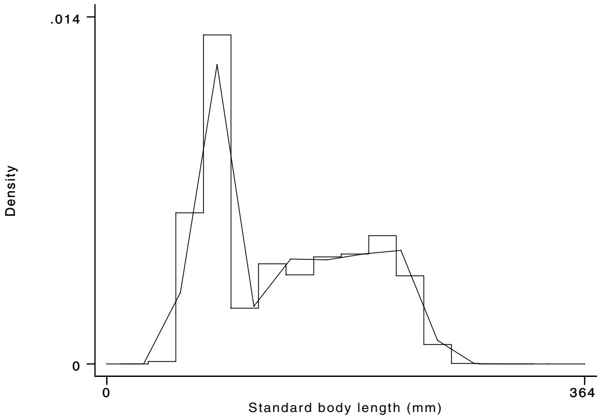

Figure 4. Oversmoothed estimates, catfish data

Following Scott and Terrell (1987) and Härdle (1991), we modified Härdle's algorithms in writing ado-files that produce WARPing versions of L2CV and BCV for kernel density estimation. These programs are called `l2cvwarp` and `bcvwarp`, respectively.

The WARPing approach is a computationally efficient method that enables us to locate the optimal bandwidth and to carry out simulations with a considerable number of observations and repetitions. To achieve this efficiency, the small bandwidth, δ, is fixed. Since $h = M \times \delta$, it is only necessary to determine the optimal value of M to find the optimal smoothing parameter. As we noted in the previous description of our Stata programs for WARPing (*snp6.1*), our ado-files rely on binary executable files (written originally in Turbo Pascal) to perform the compute-intensive portion of the WARPing procedure. This method of calling high-speed executables from within an ado-file (via Stata's `shell` command) allows us seamlessly to integrate more advanced techniques into Stata.

The syntax of `l2cvwarp` is

`l2cvwarp` *varname* [`if` *exp*] [`in` *range*] `,` `delta(`*#*`)` `kercode(`*#*`)`
[`gen(`*cvval mval hval*`)` `mend(`*#*`)` `mstart(`*#*`)` `nograph` *graph-options*]

`delta()` specifies δ, the small bandwidth resulting from the shifting of histograms to average. This value is interpreted as the measurement accuracy of the observations which are rounded to the level indicated by `delta`.

`kercode()` indicates the weight (kernel) function according to the following codes:

Code	Kernel
1	uniform
2	triangular
3	Epanechnikov
4	quartic (biweight)
5	triweight
6	Gaussian

`mstart()` and `mend()` specify the range of M considered for the minimum search. The default value for `mstart()` is 1. If `mend()` is not specified, it is set to about a third of the range of the observations.

`gen()` specifies three new variable names to contain the cross-validation, M, and h values, respectively.

`nograph` suppresses the graph.

By default, `l2cvwarp` displays a graph of the cross-validation value versus M. This graph allows you to locate visually the interval with the minimum cross-validation value. (Recall that $h = M \times \delta$.) After the graph is displayed, a table lists the five lowest CV values with the corresponding values for M and h. Thus, you can execute `l2cvwarp` iteratively to find the optimal bandwidth.

Scott (1992) and Härdle (1991) recommended displaying M and h on a semilogarithmic scale. This can be accomplished with `l2cvwarp` by adding the `xlog` option. Scott also recommended including a reference line at the value of the oversmoothed bandwidth. The `xline()` option can be used for this purpose.

We illustrate `l2cvwarp` using the snowfall data. We took a preliminary look at the data , setting $\delta = 1$ and using the default range for M (that is, we didn't specify the `mstart()` and `mend()` options). We found that the minimum value of the CV score is located in the interval $1 < M < 40$. After some trial and error, we arrived at

```
. l2cvwarp snow, delta(1) kercode(6) mstart(3) mend(30) xlog xlab xline(11.85)
(graph appears, see Figure 5)
--------------------------------------------------------------------------------
Least Squares Cross-validation for WARPing density estimation, Gaussian kernel
--------------------------------------------------------------------------------
CV-value = -0.01113733       M-value =     8     Bandwidth =   8.0000
CV-value = -0.01113101       M-value =     7     Bandwidth =   7.0000
CV-value = -0.01111789       M-value =     9     Bandwidth =   9.0000
CV-value = -0.01108621       M-value =    10     Bandwidth =  10.0000
CV-value = -0.01107764       M-value =     6     Bandwidth =   6.0000
```

Figure 5 reproduces Figure S.4.2 from Härdle's text. We have added an `xline()` indicating the oversmoothed Gaussian kernel bandwidth reported by `bandw`. Note that Härdle's estimate was obtained by using the direct algorithm, rather than the WARP approach. This difference in method accounts for the slight difference in the results (Härdle estimates that $M = 9$, compared to our estimate that $M = 8$.) As a check, we set `delta()`, `kercode()`, `mstart()`, and `mend()` to the values used by Härdle and confirmed that `l2cvwarp` produces the same values as Härdle's programs (using the updated S functions and C programs from Statlib).

Decreasing δ improves the approximation slightly, but the computational cost can easily become excessive. Moreover, it is clear that the value of the CV score is relatively insensitive to changes in h in the range from 7 to 10. For example, setting '`delta(0.5) mstar(10) mend(40)`' yields an estimate of 9.5 for the optimal bandwidth, the same as the estimate obtained by Scott (1992) in his Figure 6.16 (p. 172).

The program `bcvwarp` uses biased cross-validation to estimate the smoothing parameter. The syntax of `bcvwarp` is analogous to that of `l2cvwarp`:

`bcvwarp` *varname* [`if` *exp*] [`in` *range*] `, delta(`*#*`) kercode(`*#*`)`
[`gen(`*bcvval mval hval*`) mend(`*#*`) mstart(`*#*`) nograph` *graph-options*]

In contrast to `l2cvwarp`, `bcvwarp` supports only two type of kernel functions: quartic (`kercode(1)`) and triweight (`kercode(2)`). As we noted above, by using the conversion factors derived by Härdle (1991) and Scott (1992) and reported in Table 2 of *snp6.1*, it is possible to rescale the optimal bandwidth to correspond to any desired kernel function. In all other ways, the two programs behave similarly. `bcvwarp` displays a graph of the biased CV score against M, then lists the five lowest scores with their M and h values.

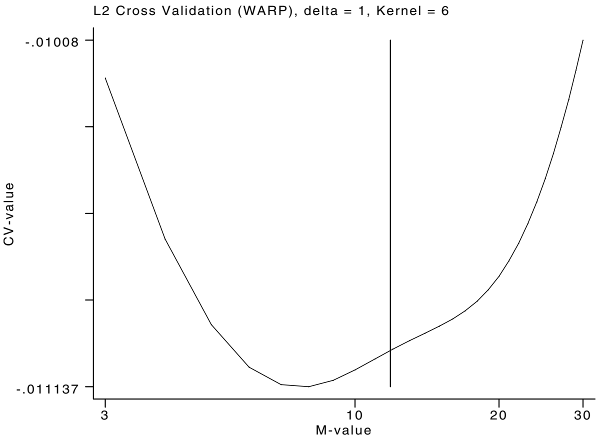

Figure 5. Least squares CV score for the snowfall data

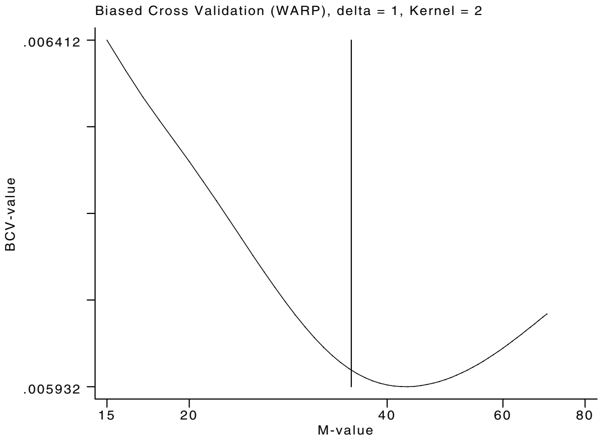

Figure 6. Biased CV score for the snowfall data

We illustrate `bcvwarp` using the snowfall data.

```
. bcvwarp snow, d(1) k(2) ms(15) me(70) xlog xlab xline(35.29)
(graph appears, see Figure 6)
--------------------------------------------------------------------------
Biased Cross-validation for WARPing density estimation, Triweight kernel
--------------------------------------------------------------------------
Biased Cv-value = 0.00593182   M-value =    43   Bandwidth =  43.0000
Biased Cv-value = 0.00593193   M-value =    42   Bandwidth =  42.0000
Biased Cv-value = 0.00593229   M-value =    44   Bandwidth =  44.0000
Biased Cv-value = 0.00593269   M-value =    41   Bandwidth =  41.0000
Biased Cv-value = 0.00593329   M-value =    45   Bandwidth =  45.0000
```

As before, we have added a reference line corresponding to the rescaled oversmoothed bandwidth reported by `bandw`. The suggested triweight kernel bandwidth is 43 which is larger than the oversmoothed h. In order to compare this estimate with the Gaussian bandwidth estimated by L2CV, we multiplied the triweight kernel bandwidth by 0.336 to obtain the optimal Gaussian bandwidth = 14.5. This result is approximately the same as the estimate reported by Scott (1992, Figure 6.16, p. 172) who employed a Gaussian BCV algorithm.

The associated density estimates are displayed in Figure 7. The smooth line displays the estimate associated with the L2CV bandwidth estimate, while the dashed line displays the estimate associated with the BCV bandwidth estimate. The L2CV smoothing parameter suggests the existence of three modes in these data. On the other hand $h(BCV)$ produces a very smooth representation without any evidence of multimodality.

As we saw earlier, the catfish data have a more complex structure than the snowfall data. We apply `l2cvwarp` and `bcvwarp` to these data here, using the quartic kernel for direct comparison.

```
. l2cvwarp bodlen, d(1) k(4) ms(5) me(25) xlog xlab xline(29)
```

(graph appears, see Figure 8)

```
------------------------------------------------------------------------------
Least Squares Cross-validation for WARPing density estimation, Quartic kernel
------------------------------------------------------------------------------
CV-value = -0.00780885       M-value =     8     Bandwidth =   8.0000
CV-value = -0.00780763       M-value =     9     Bandwidth =   9.0000
CV-value = -0.00780166       M-value =     7     Bandwidth =   7.0000
CV-value = -0.00779905       M-value =    10     Bandwidth =  10.0000
CV-value = -0.00778844       M-value =    11     Bandwidth =  11.0000
```

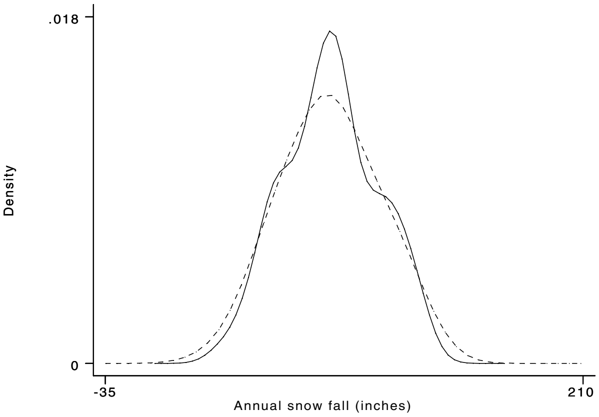

Figure 7. Density estimates using L2CV (smooth line) and BCV (dashed line) bandwidth estimates

```
. bcvwarp bodlen, d(1) k(1) ms(5) me(25) xlog xlab xline(29)
```

(graph appears, see Figure 9)

```
--------------------------------------------------------------------------
Biased Cross-validation for WARPing density estimation, Quartic kernel
--------------------------------------------------------------------------
Biased Cv-value = 0.00560286   M-value =     8   Bandwidth =   8.0000
Biased Cv-value = 0.00560392   M-value =     9   Bandwidth =   9.0000
Biased Cv-value = 0.00560413   M-value =     7   Bandwidth =   7.0000
Biased Cv-value = 0.00560660   M-value =    10   Bandwidth =  10.0000
Biased Cv-value = 0.00560979   M-value =     6   Bandwidth =   6.0000
```

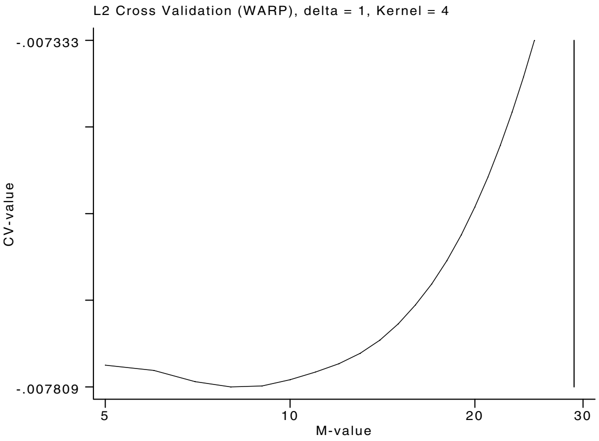

Figure 8. Least squares cross validation score for triweight kernel estimation, catfish data

Both the L2CV and BCV procedures estimate the optimal bandwidth to be 8. Figure 10 displays the quartic kernel density estimate using this estimate of the bandwidth (smooth line) and using Scott's (1992) oversmoothed estimate (dashed line). Both density estimates reveal several modes in these data. Scott has pointed out that the concordance between these different criteria represents substantial evidence that the multimodality is authentic rather than artifactual. Thus, there is a strong evidence for a multimodal distribution of the standard body length of the catfish.

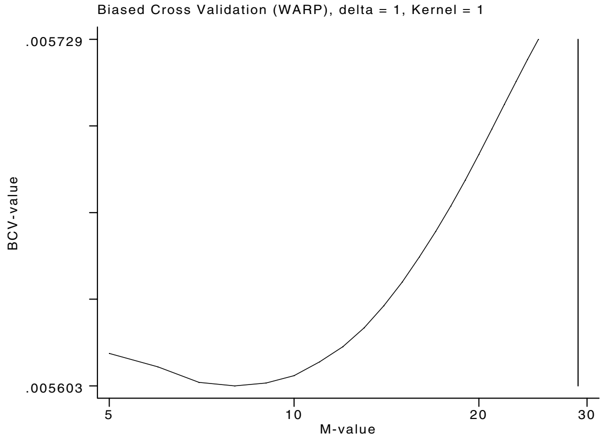

Figure 9. Biased cross validation score for triweight kernel estimation, catfish data

Other methods for choosing the smoothing parameter

Several other methods not discussed in this insert have been proposed for choosing the smoothing parameter. Two notable suggestions are bootstrap cross-validation (Taylor 1989) and plug-in methods (Sheather and Jones 1991, Hall, et al. 1991). We hope that the programs that accompany this insert motivate others to develop new commands to implement these alternative methods as well. In the meantime, the collection of rules and methods presented above provide powerful and useful insights into the selection of a smoothing parameter and, ultimately, a density estimate.

There are limitations to all these methods. As many authors have recognized (Marron 1986, Scott 1992), the practice of examining several estimates using different smoothing parameters is unlikely ever to be entirely replaced by automatic smoothing methods. From the point of view of exploratory data analysis, all bandwidth choices produce useful estimates; large h values reveal such general structural features as symmetry, outliers, modes, and location, while small h values reveal local structures which may be real or simply artifacts. Nonetheless, the search for a fully automatic and reliable bandwidth selection procedure has motivated the development of novel algorithms such as the two mentioned here. This topic remains an active area of statistical research (Scott 1992).

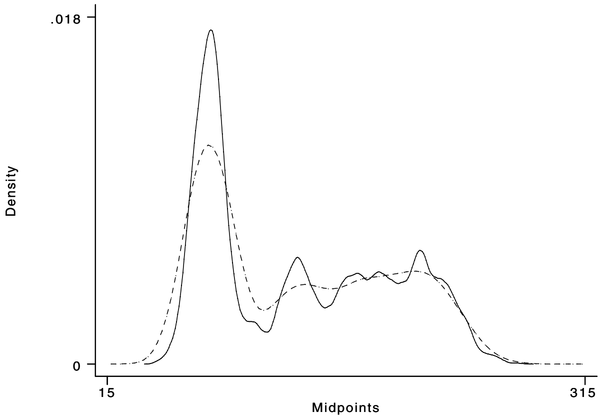

Figure 10. Density estimates using the L2CV=BCV (smooth line) and Scott's (dashed line) bandwidth estimates

A revised collection of programs for univariate density estimation

We close this insert by presenting a revised and integrated version of our programs to perform kernel density estimation by means of discretized/interpolated and ASH-WARP methods. These programs were originally introduced as a set of separate ado-files in *snp6.1*. We carried out the density estimations presented above using these new commands. The revised versions have a Stata-like syntax and include several new options. The only significant missing feature is the ability to specify weights.

As we noted in previous inserts (Salgado-Ugarte et al. 1993, 1995), kernel densities are estimated at a discrete number of grid points. In the terminology advanced by Jones (1989), both our grid-based programs and our ASH/WARP implementation belong to the classes of estimators he denoted as discretized (piecewise constant, histogram-like) or interpolated (piecewise linear, frequency-polygon-like).

These types of kernel density estimation can now be performed by `kerneld`, our revised program. The syntax of `kerneld` is

`kerneld` *varname* [`if` *exp*] [`in` *range*] `,` `bwidth(`*#*`)` `kercode(`*#*`)` `npoint(`*#*`)`
[`gen(`*denvar gridvar*`)` `nograph` *graph-options*]

`bwidth()` specifies the bandwidth, h.

`kercode()` indicates the weight (kernel) function according to the following codes:

Code	Kernel
1	uniform
2	triangular
3	Epanechnikov
4	quartic (biweight)
5	triweight
6	Gaussian
7	cosinus

`npoint()` specifies the number of equally spaced points (that is, the number of grid points) at which the kernel estimates will be calculated. Care should be taken to employ a sufficient number of points: setting `npoint()` to less than fifty points may result in a very rough estimate.

`gen()` specifies two new variable names to contain the density estimates and their associated grid points, respectively.

`nograph` suppresses the graph.

`kerneld` displays a graph of the estimated density. The graph can be modified using any of the options allowed for twoway graphs. For instance, the `connect(J)` option can be used to produce a discretized, piecewise constant estimate. However, Jones (1989) has argued that this approach may adversely affect the graphic representation. The `nograph` option will suppress the graph when you wish to use `kerneld` just to calculate and store the density estimate.

This insert also includes `warpden`, a new version integrating our programs for ASH/WARP density estimation. The syntax of `warpden` is

`warpden` *varname* [`if` *exp*] [`in` *range*] `,` `bwidth(`*#*`)` `kercode(`*#*`)` `mval(`*#*`)`
[`gen(`*denvar midvar*`)` `nosort` `step` `nograph` *graph-options*]

`bwidth()` specifies the smoothing parameter, h, which is the bin width for histograms, FPs, and averaged shifted histograms (FP-ASH) and the bandwidth for kernel density estimators.

`kercode()` indicates the weight (kernel) function using the same codes as `kerneld`. However, `warpden` does not support the cosinus kernel.

`mval()` specifies the number of shifted histograms to average.

`gen()` specifies two new variable names to contain the density estimates and the midpoints of their associated bins, respectively.

`nograph` suppresses the graph.

`nosort` indicates that the data are already sorted in the order of *varname* and omits a redundant sort operation.

`step` specifies the use of the histogram-like (ASH) estimate. If this option is omitted, the linearly interpolated (FP) estimate is used.

As before, `warpden` calls a binary executable to perform the compute-intensive portion of the estimation procedure. However, this process is now handled automatically, without prompting the user for additional information. As a consequence, this process now is invisible to the Stata user.

This insert also includes `warpdens` which has the same syntax and purpose as `warpden`. However, the "`s`" at the end of the name indicates that `warpdens` is an "all-Stata" ado-file. In other words, all the calculations are performed by Stata, and no external executable file is required. Thus, this program can be used across all Stata platforms, not just the DOS-based systems that support the executable file used by `warpden`. Of course, `warpdens` does not execute as quickly as `warpden`, especially when a high value of M is used.

Implementation notes

The programs `bandw`, `kerneld`, and `warpdens` can be used as-is by all Stata users. On the other hand, `l2cvwarp`, `bcvwarp`, and `warpden` can be used on DOS systems only. On Unix systems, the Pascal code (supplied on the distribution diskette) can be recompiled or translated to C.

Our `bcvwarp` and Härdle's analog in S do not produce the same results. The minima are located at the same value of M, but the score values are different. We have checked our code and verified that it corresponds (as far as we can tell) to the equations, algorithms, and printed programs printed in Härdle (1991). However, there are differences between these printed algorithms and the versions obtained from Statlib. It would be interesting to perform additional comparisons to resolve these discrepancies.

Despite these differences in implementation, we found that our results agree overall with those calculated by XploRe (XploRe Systems 1993).

We would appreciate hearing from users about any problems they encounter with our programs or any suggestions for improvements.

Acknowledgments

The S function and C source code from Härdle (1991) used as a basis to implement our ado-files were obtained by EMAIL from Statlib. Dr. Brian Ripley kindly sent listings of the results obtained with S for comparison and validation. With the assistance of Dr. J. Hilbe and Dr. W. Härdle, it was possible to use the XploRe program for additional checking. The first author is grateful to the Ministry of Education, Science and Culture of Japan and to the Universidad Nacional Autonomous University of Mexico (FES Zaragoza and DGAPA) for their support.

References

Bowman, A. W. 1984. An alternative method of cross-validation for the smoothing of density estimates. *Biometrika* 71: 353–360.

Chiu, S. T. 1991. Bandwidth selection for kernel density estimation. *Annals of Statistics* 19: 1883–1905.

Comparini, A. and E. Gori. 1986. Estimating modes and antimodes of multimodal densities. *Metron* (Italian Statistical Review) 44: 307–332.

Doane, D. P. 1976. Aesthetic frequency classifications. *The American Statistician* 30: 181–183.

Fisher, R. A. 1932. *Statistical methods for research workers*. 4th ed. Edinburgh: Oliver and Boyd.

——. 1958. *Statistical methods for research workers*. 13th ed. Edinburgh: Oliver and Boyd.

Emerson J. D. and D. C. Hoaglin. 1983. Stem-and-leaf displays. In *Understanding robust and exploratory data analysis*. ed. Hoaglin, D. C., F. Mosteller and J. W. Tukey. New York: John Wiley & Sons, 7–30.

Freedman, D. and P. Diaconis. 1981a. On the histogram as a density estimator: L theory. *Zeitschrift fur Wahrscheinlichkeitstheorie und verwandte Gebiete* 57: 453–476.

——. 1981b. On the maximum deviation between the histogram and the underlying density. *Zeitschrift fur Wahrscheinlichkeitstheorie und verwandte Gebiete* 58: 139–167.

Frigge, M., D. C. Hoaglin, and B. Iglewicz. 1989. Some implementations of the boxplot. *The American Statistician* 43: 50–54.

Hall P., S. J. Sheather, M. C. Jones, and J. S. Marron. 1991. On optimal data-based bandwidth selection in kernel density estimation. *Biometrika* 27: 228–254.

Hamilton, L. C. 1992. sed6: Quartiles, outliers, and normality: some Monte Carlo results. *Stata Technical Bulletin* 6: 4–5.

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

Hoaglin, D. C. 1983. Letter values: A set of selected order statistics. In *Understanding robust and exploratory data analysis*. ed. Hoaglin, D. C., F. Mosteller and J. W. Tukey. New York: John Wiley & Sons, 33–57.

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

Jones, M. C. 1989. Discretized and interpolated kernel density estimates. *Journal of the American Statistical Association* 84: 733–741.

Marron, J. S. 1986. Will the art of smoothing ever become a science? *Contemporary Mathematics* 59: 169–178.

Müller, D. W. and G. Sawitzki. 1991. Excess mass estimates and tests for multimodality. *Journal of the American Statistical Association* 86: 738–746.

Parzen, E. Nonparametric statistical data modeling. *Journal of the American Statistical Association* 74: 105–131.

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

Rudemo, M. 1982. Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics* 9: 65–78.

Salgado-Ugarte, I. H. 1985. Algunos aspectos biológicos del bagre *Arius melanopus* Gunther (Osteichthyes: Ariidae) en el Sistema Lagunar de Tampamachoco, Ver. B. S. thesis, Carrera de Biología, E.N.E.P. Zaragoza, Universidad Nacional Autónoma de México.

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

——. 1995. snp6.1: ASH, WARPing, and kernel density estimation for univariate data. *Stata Technical Bulletin* 26: 23–31.

Scott, D. W. 1979. On optimal and data-based histograms. *Biometrika* 66: 605–610.

——. 1985a. Averaged shifted histograms: effective nonparametric density estimators in several dimensions. Annals of Statistics 13: 1024–1040.

——. 1985b. Frequency polygons: Theory and application. *Journal of the American Statistical Association* 80: 348–354.

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

Scott, D. W. and G. R. Terrell. 1987. Biased and unbiased cross-validation in density estimation. *Journal of the American Statistical Association* 82: 1131–1146.

Sheather, S. J. and M. C. Jones. 1991. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society, B* 53: 683–690.

Silverman, B. W. 1978. Choosing the window width when estimating a density. *Biometrika* 65: 1–11.

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

——. 1983. Some properties of a test for multimodality based on kernel density estimates. In *Probability, Statistics and Analysis*. ed. Kingman, J. F. C. and G. E. H. Reuter, 248–259. Cambridge: Cambridge University Press, 248–259.

——. 1986. *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall.

Stata Corporation. 1995. *Reference Manual, Release 4* College Station, TX: Stata Press.

Sturges, H. A. 1926. The choice of a class interval. *Journal of the American Statistical Association* 21: 65–66.

Taylor, C. C. 1989. Bootstrap choice of the smoothing parameter in kernel density estimation. *Biometrika* 76: 705–712.

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

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

Tukey, J. W. 1977. *Exploratory Data Analysis*. Reading: Addison–Wesley.

XploRe Systems. 1993. *XploRe—A computing Environment for eXploratory Regression and Data Analysis. Version 3.1* Institut für Statistik und Ökonometrie. Berlin.

ssa7 | Analysis of follow-up studies

David Clayton, MRC Biostatistical Research Unit, Cambridge, EMAIL david.clayton@mrc-bsu.cam.ac.uk
Michael Hills, London School of Hygiene and Tropical Medicine, London, EMAIL mhills@lshtm.ac.uk

This entry describes three commands for analyzing follow-up studies using the simple tabulation and stratification methods described in Part I of Clayton and Hills (1993). The first command, `lexis`, divides the follow-up time for each subject into single bands of time during which the rate is assumed to be constant. Once this has been done, `tabrate` can be used to tabulate and graph rates, and `mhrate` can be used to find Mantel–Haenszel estimates of rate ratios controlled for one or more confounding variables. The `tabrate` and `mhrate` commands, together with the equivalent commands for odds described in a companion entry (*ssa8*), are alternatives to the commands in `epitab` [5s].

Subdivision of follow-up time by bands

The command lexis divides the follow-up for each record into bands, using a variable which is the time of entry to the follow-up study on a particular time-scale, such as age, or calendar period, or time in study. Each band of this time scale occupies one new record, so the data set is expanded. Expansion on several time scales can be achieved by repeated calls to lexis. The syntax of lexis is

lexis *timein fail fup* , breaks($x_1, x_2, \ldots, x_k$) generate(*varname*) [update(*varlist*)]

Before expansion, the variable *fup* contains the total follow-up time, and the variable *fail* contains the outcome at exit. The outcome should be coded to show the type of failure, with '0' for censored observations. The variable *timein* contains the entry time on the time scale on which the observations are being expanded. After expansion, *timein* and *fail* contain the entry time and follow-up time for the bands defined in the option breaks(). The number of new records created is shown. Since the current data set will be altered by this command, if and in options are best implemented using keep or drop.

This command adds a new variable, _lexis, to the data set. This variable contains the observation numbers of the original data set before any expansion. _lexis is used to check that time variables in later calls to lexis have been updated.

Options

breaks($x_1, x_2, \ldots, x_k$) is not optional. It defines the break points for the bands. The first and last break points define the span of the study, according to the following rules. Records for which the time of exit from the study is less than the first break point are dropped. Similarly records for which the time of entry is greater than the last break point are dropped. Otherwise, the time of entry is redefined as the larger of time at entry and the first break point, and the time of exit is redefined as the smaller of time at entry and the largest break. Finally, for records in which the time of exit is greater than the largest break point, the failure indicator is set to zero (censored), no matter what its original value.

generate() is not optional. This option supplies the name of a new categorical variable to hold the time bands, coded using the left-hand ends of the intervals defined by breaks().

update() specifies variables which contain entry times on other time scales. These variables will be appropriately incremented, thus allowing their use in further calls to lexis.

Background

During a long follow-up, the rates of morbidity and mortality experienced by a cohort may change. The standard method of analysis in this situation is to divide the observation time for each subject into bands of time during which the rates are considered to be constant. The study of events through time is greatly helped by the use of the lexis diagram which is why we have called the new command lexis.

We illustrate the use of lexis with data from a heart disease and diet survey. These data are included with this submission as the file kcal.dta. The data arose from a study described more fully in Morris, Marr, and Clayton (1977), and they are analyzed in Clayton and Hills. Note, however, that the results given there differ slightly from those given here, due to updating the file.

```
. use kcal,clear
(Heart disease and diet survey)

. describe

Contains data from kcal.dta
  Obs:   337 (max= 51799)                 Heart disease and diet survey
 Vars:    10 (max=    99)
Width:    26 (max=   200)
  1. id           int    %8.0g            identity number
  2. agein        float  %9.0g            age at entry
  3. y            float  %9.0g            person-years
  4. d            byte   %8.0g            ihd deaths
  5. job          byte   %8.0g            driver/conductor/bank
  6. month        byte   %8.0g            month of survey
  7. loweng       byte   %8.0g            low energy
  8. toteng       float  %9.0g            total energy (kcals/day)
  9. height       float  %9.0g            height(cms)
 10. weight       float  %9.0g            weight(kgs)
```

The variable agein contains the age at entry to the study, y contains the time the subject spends in the study, and d contains the outcome, coded one for a death from heart disease and zero otherwise (i.e., censored). Other variables, such as height, do not vary with time. A listing of some of the variables for two subjects is given below.

```
. list agein y d loweng height if id==1, noobs

    agein          y         d    loweng     height
    49.62      12.29         0         0    175.514

. list agein y d loweng height if id==34, noobs

    agein          y         d    loweng     height
    59.84   7.710003         1         1      177.8
```

The variable `loweng` is coded one if the total energy consumption is low (<2750 Kcals) and zero otherwise; it is the main explanatory variable of interest in this study. Subject 1, who has normal energy consumption, enters at age 49.62 and exits 12.29 years later, when his follow-up is censored. Similarly subject 34, who has low energy consumption, enters at age 59.84 and exits 7.71 years later when he dies from heart disease.

Example 1: Age

In order to control for actual age (as opposed to age at the start of the study), it is necessary first to expand the data so that each new record refers to the observation of a subject through a single age band for which the rate is assumed to be constant. Using ten-year age bands starting at age 40, and finishing at age 70, the `lexis` command takes the form

```
. lexis agein d y, generate(ageband) breaks(40,50,60,70)
26   records start before first break - left censored
392  extra records created

NOTE: Following lexis expansion on agein
the following variables have been updated: agein
```

The output indicates that 26 subjects are less than 40 years of age at entry, and their follow-up has been left censored, that is, their age at entry has been replaced by 40. The number of new records created acts as a warning that the data set has changed radically. The note about which variables have been updated is useful when more than one time scale is being considered.

The effect of this command on the data is shown by

```
. list agein y d ageband height if id==1, noobs

    agein          y         d   ageband     height
    49.62   .3800011         0        40    175.514
       50         10         0        50    175.514
       60       1.91         0        60    175.514

. list agein y d ageband height if id==34, noobs

    agein          y         d   ageband     height
    59.84   .1599998         0        50      177.8
       60   7.550003         1        60      177.8
```

This example shows how the single record for subject with `id==1` has expanded to three records. The first refers to the age band 40–49, coded 40, and the subject spends 0.38 years in this band. The second refers to the age band 50–59, coded 50, and the subject spends 10 years in this band. Finally the third refers to the age band 60–69, coded 60, and the subject spends 1.91 years in this band. The follow-up in each of the three bands is censored (`d==0`). The single record for the subject with `id==34` is expanded to two age bands; the follow-up for the first band is censored (`d==0`) and the follow-up for the second band ends in death (`d==1`).

The values for variables which do not change with time, such as `height`, are simply repeated in the new records. This can lead to much larger data sets after expansion, and it may be necessary to drop any unneeded variables before using `lexis`. Once the records have been expanded, they can be analyzed as if they came from separate independent subjects, using simple tabulation, stratification with Mantel–Haenszel estimates, or `poisson` regression.

Example 2: Time in study

It is also possible to use `lexis` to expand the records by (say) five-year bands of time in study. First we need to clear the current data and to create a `timein` variable which takes the value zero for all subjects:

```
. use kcal, clear
(Heart disease and diet survey)

. gen timein=0
. lexis timein d y, generate(timeband) breaks(0,5,10,15,20,25)
767  extra records created

NOTE: Following lexis expansion on timein
the following variables have been updated: timein

. list timein d y timeband height if id==1, noobs

   timein         d          y  timeband     height
        0         0          5         0    175.514
        5         0          5         5    175.514
       10         0   2.290001        10    175.514
```

The subject with `id==1` spends five years in the time band 0–4, five years in 5–9, and 2.29 years in 10–14. Follow-up in each of these bands is censored.

Example 3: Age and time in study

The command `lexis` can be used to expand the records on two time scales, such as age and time in study. To do this we need first to expand on the age scale and then on the time in study scale. The `lexis` command can be used sequentially in this way, but, when expanding by age, the `timein` variable also must be updated to refer to the new records. This is done with the option `update`, as follows:

```
. use kcal, clear
(Heart disease and diet survey)

. gen timein=0

. lexis agein d y, generate(ageband) breaks(40,50,60,70) update(timein)
26   records start before first break - left censored
392  extra records created

NOTE: Following lexis expansion on agein
the following variables have been updated: agein timein

. list ageband agein timein y d if id==1, noobs

 ageband      agein     timein          y         d
      40      49.62          0   .3800011         0
      50         50   .3800011         10         0
      60         60      10.38       1.91         0

. lexis timein d y, generate(timeband) breaks(0,5,10,15,20,25) update(agein)
761  extra records created

NOTE: Following lexis expansion on timein
the following variables have been updated: timein agein

. list ageband agein timeband timein y d if id==1, noobs

 ageband      agein  timeband     timein          y         d
      40      49.62         0          0   .3800011         0
      50         50         0   .3800011   4.619999         0
      50      54.62         5          5          5         0
      50      59.62        10         10   .3800011         0
      60         60        10      10.38       1.91         0
```

Note how, after the first use of `lexis`, `timein` has been updated from zero to the time at which the subject enters each age band. Similarly `agein` has been updated, after the second use of `lexis`, to refer to the age at which the subject enters each band of time in study.

Example 4: Explanatory variables that change with time

In the previous examples, time itself, in the shape of age or time in study, is the explanatory variable which is to be studied or controlled for, but in some studies there are other explanatory variables which vary with time. The `lexis` command can sometimes be used to expand the records so that in each new record such an explanatory variable is constant over time. For example, in a study of the effect of bereavement on mortality, (Jagger and Sutton 1991), elderly married couples were followed in time. Initially both members of a couple were alive, so neither was bereaved, but after the death of one the other became bereaved. Some typical records might look like this:

```
 index      entry       exit   death.sp
    60       7690       7885       8035
    63       7690       8035       7885
   156       7690      11323      10554
   220       7690      10554          .
```

The first two records refer to a couple who entered at 7690 days (elapsed days from January 1, 1960); one died at 7885 days, thus bereaving the other, the other died at 8035 days. The variable `death.sp` refers to the time of death of the subject's spouse. In the next pair of records, both subjects again entered at 7690 days; one died at 10554 days, thus bereaving the other, the other was still alive at the end of follow-up. To study the effect of bereavement, the second and fourth records need to be expanded into unbereaved and bereaved parts.

This expansion can be done by creating a new time scale on which time before bereavement (or before exit, if not bereaved) takes negative values, and time since bereavement takes positive values. For example subject 60 spends $7690 - 7885 = -195$ days before exit, while subject 63 spends $7690 - 7885 = -195$ days before bereavement and $8035 - 7885 = 150$ days after bereavement. Thus, the new time scale is

```
. generate newtime=cond(death.sp>exit, entry-exit, entry-death.sp)
```

The follow-up on this scale is then split using the breaks `(-10000, 0, 10000)`. The new variable which holds the left-hand end of the bands should be recoded so that $-1000 = 0$, $0 = 1$. If the first year of bereavement is of particular interest, the breaks `(-10000, 0, 365, 10000)` could be used.

The same approach can be taken to the Stanford heart data, described in the *Stata Reference Manual* ([5s] cox), by setting the origin of the time scale to the day of the heart transplant, if this happened, and to the day of exit from the study otherwise.

Different types of records

The data for follow-up studies usually start as individual records, where each subject in the study has his or her own record. These may then be expanded by age or other time scales. For large studies, records are often aggregated by summing the number of failures and the observation time over records with the same values for a group of explanatory variables. This is done by collapsing over `fail` and `y`, as in

`collapse fail y, sum(D Y) by(`*varlist*`)`

Each new record now contains D and Y, the total failures and total observation time, for each combination of values of the variables in the *varlist*. These records are closely related to frequency records, and we shall refer to them as Poisson frequency records.

Tabulating the rate

Simple tabulation of rates can be carried out fairly easily using the Stata `collapse` command, from either individual records or from Poisson frequency records, but, because it is a frequent operation during data exploration, we have created a new command, `tabrate`, with syntax

`tabrate` *fail* [*xvar*] [`in` *range*] [`if` *exp*] , [`exposure(`*fup*`)` `graph` `level(`*#*`)` `smr` `trend`]

where *xvar* is a categorical explanatory variable. If *xvar* is absent, the overall rate is reported.

Options

`exposure()` supplies the follow-up time for rates or the expected numbers of cases for standardized mortality ratios (SMRs). The use of the term "exposure" comes from situations in which the observation time is also the length of time for which a subject was exposed to risk. It should not be confused with the more common use of the term in epidemiology to refer to an exogenous explanatory variable. A less ambiguous term, used in demography, is *rate multiplier*, so called because the rate (or SMR) multiplied by this quantity yields the number of events, but in this submission we have used the term exposure for compatibility with the `poisson` command.

`graph` produces a graph of the rate against the numerical code used for the categories of *xvar*. Graph options other than `connect()` are allowed.

`level()` gives the level for the confidence intervals.

`smr` labels the output more suitably for SMRs. The default is for rates.

`trend` produces a test (the score test) for a linear trend of the log rate against the numerical code used for the categories of *xvar*. In the absence of `trend`, an approximate χ^2 test for unequal rates (heterogeneity) is produced.

Example 5

After expanding on the age scale using `breaks(40,50,60,70)`, the data set consists of 729 records. The mortality rate can now be tabulated against ageband using the command

```
. tabrate d ageband, exposure(y)
table of cases (D), person-years (Y), and rates per 1000 person-years

 ageband        _D         _Y     _rate    ci_low   ci_high
      40         6      907.0     6.615     2.972    14.725
      50        18     2107.0     8.543     5.382    13.559
      60        22     1493.4    14.732     9.700    22.374

Chisq test for unequal rates =     4.71 (2 df, p =  0.095 )
```

If the mortality rate is tabulated by a variable which does not change with time, such as `loweng`, the result is exactly the same as it would have been before expansion, provided no records were dropped in expansion.

Standardized Mortality Ratios (SMRs)

The SMR for a cohort is the ratio of the total number of observed deaths to the number expected from age-specific reference rates. This expected number can be found by, first, expanding on age, using `lexis`, and then multiplying the person years in each age band by the reference rate for that band. The Stata command `merge` can be used to add the reference rates to the data set, ready for multiplication by the person-years. A double expansion on age and calendar period can be used to produce expected numbers from age × period-specific reference rates.

Rate ratios

The command `mhrate` is useful for estimating rate ratios, controlled for confounding using stratification. It is similar to the command `ir` from the `epitab` suite of commands, but with more flexibility and less output. The syntax of `mhrate` is

`mhrate` *fail xvar* [*varlist*] [`if` *exp*] [`in` *range*] ,

`exposure(`*varname*`)` [`by(`*varlist*`)` `compare(`v_1, v_2`)` `level(`*#*`)`]

`mhrate` estimates the ratio of the rates of failure for two categories of *xvar*, controlled for specified confounding variables, and also tests whether this rate ratio is equal to one. When *xvar* has more than two categories but none are specified in a `compare` option, `mhrate` assumes *xvar* is quantitative and calculates a one-degree-of-freedom test for trend. It also calculates an approximate estimate of the rate ratio for a one unit increase in *xvar*. This is a one-step Newton–Raphson approximation to the maximum likelihood estimate and is equal to the ratio of the score statistic, U, to its variance, V (Clayton and Hills, p. 103).

The variable *fail* indicates failure (1) or censoring (0) after the follow-up time supplied by `exposure()`. Alternatively, the command may be used with records where the failure variable contains the total number of failures for each combination of explanatory variables and `exposure()` supplies the corresponding total person-years observation. The remaining variables are categorical variables that are to be controlled for using stratification. Strata are defined by cross-classification of all these variables and the rate ratio estimate is combined over strata using the Mantel–Haenszel method. Confidence intervals are calculated for the rate ratio using the formula $V/(QR)$ for the variance of the log of the MH estimate (Clayton and Hills, p. 146). In those circumstances where a trend test is calculated, the 1-d.f. test corresponds to the Mantel–Haenszel extension test (Clayton and Hills, p. 203).

Using the `by()` option, the variation of rate ratios with further categories may be explored. When this option is used, a Mantel–Haenszel combined estimate and a test for unequal rate ratios are also computed.

Options

`exposure( )` is not optional. It specifies the variable that contains the person-years of observation time (expected numbers with SMRs).

`by( )` specifies categorical variables by which the rate ratio is to be tabulated. A separate rate ratio is produced for each category or combination of categories, and a test for unequal separate rate ratios is performed.

compare(v_1, v_2) specifies the categories of *xvar* to be compared; v_1 defines the numerator and v_2 the denominator. When compare is absent and there are only two categories, the second is compared to the first; when there are more than two categories, an approximate estimate of the rate ratio for a unit increase in *xvar*, controlled for specified confounding variables, is given.

level( ) gives the level for the confidence intervals.

Example 6

After expanding the records using lexis on the age scale, the rate ratio for loweng, level 1 compared to level 0, controlled for ageband, can be found using

```
. mhrate d loweng ageband, exposure(y) compare(1,0)
Mantel-Haenszel estimate of the rate ratio
Comparing loweng==1 vs loweng==0, controlling for ageband
RR estimate, lower and upper 95% confidence limits, and
chi-squared test for RR=1 (1 degree of freedom)
    RR   Lower   Upper   Chisq  p_value
 1.873   1.029   3.409   4.357   0.037
```

Example 7

To make the same comparison separately by job, try

```
. mhrate d loweng ageband, exposure(y) compare(1,0) by(job)
Mantel-Haenszel estimate of the rate ratio
Comparing loweng==1 vs loweng==0, controlling for ageband
by job
RR estimate, lower and upper 95% confidence limits, and
chi-squared test for RR=1 (1 degree of freedom)
     job      RR   Lower   Upper   Chisq  p_value
       0   2.393   0.750   7.634   2.313   0.128
       1   1.564   0.534   4.582   0.676   0.411
       2   1.945   0.796   4.752   2.210   0.137
Mantel-Haenszel estimate controlling for: ageband job
    RR   Lower   Upper   Chisq  p_value
 1.921   1.065   3.463   4.882   0.027
Approx chisq for unequal RRs (effect modification)    0.28 (2 df, p = 0.869)
```

Example 8

This example illustrates what happens when *xvar* is a quantitative variable, in this case ht5, the result of grouping height in 8 cm intervals from 152 to 192.

```
. sort height
. gen int ht5=autocode(height,5,152,192)
(10 missing values generated)
. replace ht5=ht5-4
(719 real changes made)
. tabulate ht5
        ht5|      Freq.     Percent        Cum.
------------+-----------------------------------
        156 |         10        1.39        1.39
        164 |        138       19.19       20.58
        172 |        295       41.03       61.61
        180 |        236       32.82       94.44
        188 |         40        5.56      100.00
------------+-----------------------------------
      Total |        719      100.00
```

mhrate now tests for a trend of heart disease rates with height, and also provides a rough estimate of the rate ratio for a 1 cm increase in height.

```
. mhrate d ht5 , exposure(y)
Score test for trend of rates with ht5
RR estimate, lower and upper 95% confidence limits, and
chi-squared test for trend (1 degree of freedom)
The RR estimate is an approximate estimate of the
```

```
rate ratio for one unit increase in ht5

    RR   Lower   Upper   Chisq  p_value
 0.925   0.887   0.964  13.381   0.000
```

There is clear evidence for decreasing rate with increasing height. The next command further examines this finding by occupational group.

```
. mhrate d ht5 , exposure(y) by(job)

Score test for trend of rates with ht5
by job

RR estimate, lower and upper 95% confidence limits, and
chi-squared test for trend (1 degree of freedom)

The RR estimate is an approximate estimate of the
rate ratio for one unit increase in ht5

     job      RR   Lower   Upper   Chisq  p_value
       0   1.015   0.925   1.113   0.093   0.761
       1   0.909   0.821   1.007   3.333   0.068
       2   0.873   0.815   0.935  15.214   0.000

Mantel-Haenszel estimate controlling for:  job

    RR   Lower   Upper   Chisq  p_value
 0.918   0.874   0.963  12.059   0.000

Approx chisq for unequal RRs (effect modification)    6.58 (2 df, p = 0.037)
```

Note that since the RR estimates are approximate, the test for their equality is also approximate.

References

Clayton, D. G. and M. Hills. 1993. *Statistical Models in Epidemiology*. Oxford: Oxford University Press.

Jagger, C. and C. J. Sutton. 1977. Death after marital bereavement—is the risk increased? *Statistics in Medicine* 10: 395–404.

Morris, J. N., J. W. Marr, and D. G. Clayton. 1977. Diet and heart: a postscript. *British Medical Journal* 19: 1307–14.

ssa8 Analysis of case–control and prevalence studies

David Clayton, MRC Biostatistical Research Unit, Cambridge, EMAIL david.clayton@mrc-bsu.cam.ac.uk
Michael Hills, London School of Hygiene and Tropical Medicine, London, EMAIL mhills@lshtm.ac.uk

This entry describes two commands for analyzing case–control and prevalence studies using the simple tabulation and stratification methods described in Part I of Clayton and Hills (1993). The command `tabodds` tabulates the odds against the numerical codes used in a categorical variable, with `graph` and `trend` options, while `mhodds` produces Mantel–Haenszel estimates of odds ratios, combined over strata. The `tabodds` and `mhodds` commands, together with equivalent commands for rates, described in a companion submission (*ssa7*), are alternatives to the commands in `epitab` [5s].

The outcome for each subject is whether the subject is a case or a control in a case–control study, or whether the subject exhibits the disease or not in a prevalence study. Being a case or having the disease is coded 1, and we shall refer to this outcome as failure; the other outcome is coded 0. The outcome variable will be called `fail`. This terminology ties in with that used for follow-up studies, and, indeed, when a five-year follow-up study is analyzed using risk (cumulative incidence), the variable `fail` records whether or not a subject fails during the five years, and the commands described here can be used to compare the risk between groups.

Different types of records

The data for case–control and prevalence studies can be arranged in three different ways. Most common is individual records, where each subject in the study has his or her own record. Closely related are frequency records where identical individual records are included only once, but with a variable giving the frequency with which the record occurs. Frequency records can be obtained from individual records by generating a variable `one` which takes the value 1 for all individuals, and using

```
. collapse one, sum(freq) by(varlist)
```

where *varlist* refers to all the other variables in the individual record. Similarly individual records can be obtained from frequency records using ‘`expand freq`’. In the third type of coding, identical individual records are further aggregated by collapsing over `fail` as well as over `one`, as in

```
        collapse fail one, sum(D N) by(varlist)
```

Each record then contains D, the number of failures out of N subjects, together with the other variables. For convenience we shall refer to this type of record as a binomial frequency record.

Tabulating the odds of failure

The command `tabodds` has syntax

`tabodds` *fail* [*xvar*] [`if` *exp*] [`in` *range*] [*fweight*] , [`binomial(`*varname*`)` `graph` `level(`*#*`)` `trend`]

`tabodds` tabulates the odds of failure against a categorical explanatory variable *xvar*. When *xvar* is absent, the overall odds is reported. The variable *fail* is coded 0/1 for individual and simple frequency records, and equals the number of failures for binomial frequency records.

Options

`binomial()` supplies the number of subjects for binomial frequency records. For individual and simple frequency records this option is not used.

`graph` produces a graph of the odds against the numerical code used for the categories of *xvar*. Graph options other than `connect()` are allowed.

`level()` gives the level for the confidence intervals.

`trend` produces a (score) test for a linear trend of the log odds against the numerical code used for the categories of *xvar*. In the absence of `trend`, an approximate χ^2 test for unequal odds is produced.

Odds ratios

The command `mhodds` has syntax

`mhodds` *fail xvar* [*varlist*] [`if` *exp*] [`in` *range*] [*fweight*] ,
[`binomial(`*varname*`)` `by(`*varlist*`)` `compare(`v_1, v_2`)` `level(`*#*`)`]

`mhodds` estimates the ratio of the odds of failure for two categories of *xvar*, controlled for specified confounding variables, and also tests whether this odds ratio is equal to one. When *xvar* has more than two categories but none are specified in a `compare` option, `mhrate` assumes *xvar* to be a quantitative variable and calculates a one-degree of freedom test for trend. It also calculates an approximate estimate of the rate ratio for a one unit increase in *xvar*. This is a one-step Newton–Raphson approximation to the maximum-likelihood estimate calculated as the ratio of the score statistic, U, to its variance, V (Clayton and Hills, 1993, p. 103).

The variable *fail* is coded 0/1 for individual and simple frequency records and equals the number of failures for binomial frequency records. The remaining variables preceding the options are categorical variables that are to be controlled for using stratification. Strata are defined by cross-classification of all of these variables, and the odds ratio estimate is combined over strata using the Mantel–Haenszel method. Using the `by` option, the variation of the combined odds ratio with further categorical variables can be explored. The formula used for the variance of the Mantel–Haenszel estimate is the one given in Clayton and Hills (p. 178). This simple formula has been justified by Martyn Plummer (1995).

A warning message is printed if some of the strata in the Mantel–Haenszel estimate of the effect of *xvar* make no contribution to the estimate. This is a useful warning that you may have over-stratified, particularly in matched studies.

Options

`binomial()` supplies the number of subjects for binomial frequency records. This option is not used for either individual or simple frequency records.

`by( )` specifies categorical variables by which the odds ratio is to be tabulated. A separate odds ratio is produced for each category or combination of categories, and a test whether these separate odds ratios are unequal is given. The same treatment applies to the odds ratio for a unit increase in *xvar*.

`compare(`v_1, v_2`)` gives the categories of *xvar* to be compared; v_1 defines the numerator and v_2 the denominator. When `compare` is absent and there are only two categories, the second is compared to the first; when there are more than two categories an approximate estimate of the odds ratio for a unit increase in *xvar*, controlled for specified confounding variables, is given.

`level()` gives the level for the confidence intervals.

Example

We illustrate the use of `tabodds` and `mhodds` with the data from the Ille-et-Villaine study of oesophageal cancer discussed in Breslow and Day (1980, chapter 5). The data are in the form of binomial frequency records in which `D` is the number of cases and `H` the number of (healthy) controls for each combination of six age-groups, four levels of alcohol, and four levels of tobacco. The derived variable `N` is the sum of `D` and `H`. The command `tabodds` can be used to tabulate the odds against a single categorical variable, for example

```
. use bdiev

. describe

Contains data from bdiev.dta
  Obs:    88 (max= 30488)
 Vars:     5 (max=    99)
Width:     7 (max=   200)
  1. age          byte   %8.0g                age in 10 year grps
  2. alc          byte   %8.0g                alcohol
  3. tob          byte   %8.0g                tobacco
  4. cases        int    %8.0g                cases
  5. controls     int    %8.0g                controls
Sorted by:

. generate N=cases+controls

. generate D=cases

. tabodds D age, binomial(N)

table of cases (D), controls (H), and odds (D/H)

     age         _D         _H     _odds    ci_low   ci_high
       1          1        115     0.009     0.001     0.062
       2          9        190     0.047     0.024     0.092
       3         46        167     0.275     0.199     0.382
       4         76        166     0.458     0.349     0.601
       5         55        106     0.519     0.375     0.719
       6         13         31     0.419     0.219     0.801

Chisq test for unequal odds =    96.94 (5 df, p =  0.000 )
```

shows age to be potentially a strong confounder. Graph options can be used to study the shape of the relationship of the odds with age. Similarly

```
. tabodds D alc, binomial(N)

table of cases (D), controls (H), and odds (D/H)

     alc         _D         _H     _odds    ci_low   ci_high
       1         29        386     0.075     0.052     0.110
       2         75        280     0.268     0.208     0.346
       3         51         87     0.586     0.415     0.828
       4         45         22     2.045     1.228     3.406

Chisq test for unequal odds =   158.79 (3 df, p =  0.000 )
```

shows a steady increase in odds with alcohol consumption. The first use of `mhodds` is to estimate the effect of alcohol controlled for age, and while we are at it we may as well do this by levels of tobacco consumption.

```
. mhodds D alc age, binomial(N) by(tob)

Score test for trend of odds with alc
controlling for age
by tob
WARNING: only 19 of the 24 strata formed in this analysis
contribute information about the effect of the explanatory variable

OR estimate, lower and upper 95% confidence limits, and
chi-squared test for OR=1 (1 degree of freedom)
(The OR estimate is an approximation to the odds ratio
for one unit increase in alc)

     tob      OR   Lower   Upper   Chisq  p_value
       1   3.580   2.687   4.769  75.949   0.000
       2   2.304   1.669   3.179  25.773   0.000
       3   2.364   1.488   3.756  13.271   0.000
       4   2.218   1.312   3.750   8.839   0.003
Mantel-Haenszel estimate controlling for: age tob
    OR   Lower   Upper   Chisq  p_value
 2.751   2.293   3.301  118.370   0.000
Approx chisq for unequal ORs (effect modification)    5.46 (3 df, p = 0.141)
```

The results show an effect of alcohol, controlled for age, of about ×2.7, which is consistent across different levels of tobacco consumption. Similarly

```
. mhodds D tob age, binomial(N) by(alc)
Score test for trend of odds with tob
controlling for age
by alc
WARNING: only 18 of the 24 strata formed in this analysis
contribute information about the effect of the explanatory variable
OR estimate, lower and upper 95% confidence limits, and
chi-squared test for OR=1 (1 degree of freedom)
(The OR estimate is an approximation to the odds ratio
for one unit increase in tob)
     alc      OR   Lower   Upper   Chisq  p_value
       1   2.421   1.561   3.753  15.608   0.000
       2   1.428   1.067   1.910   5.749   0.016
       3   1.472   0.975   2.223   3.381   0.066
       4   1.215   0.739   1.998   0.588   0.443
Mantel-Haenszel estimate controlling for: age alc
    OR   Lower   Upper   Chisq  p_value
 1.553   1.281   1.884  20.070   0.000
Approx chisq for unequal ORs (effect modification)    5.26 (3 df, p = 0.154)
```

shows an effect of tobacco, controlled for age, of about ×1.5, which is consistent across different levels of alcohol consumption. Comparisons between particular levels of alcohol and tobacco consumption can be made by generating a new variable with levels corresponding to all combinations of alcohol and tobacco, as in

```
. egen alctob=group(alc tob)
. mhodds D alctob, binomial(N) compare(16,1)
Maximum likelihood estimate of the odds ratio
Comparing alctob==16 vs alctob==1
OR estimate, lower and upper 95% confidence limits, and
chi-squared test for OR=1 (1 degree of freedom)
    OR   Lower   Upper   Chisq  p_value
93.333  14.766  589.938  103.212   0.000
```

which shows an odds ratio of 93 between subjects with the highest levels of alcohol and tobacco, and those with the lowest levels.

Matched case–control studies

Matched case–control studies, where cases and controls in each matched set share common values of the matching variables, can be analyzed using `mhodds` by controlling on the variable used to identify the matched sets. For example, when the variable `set` is used to identify which matched set each subject is in,

```
. mhodds fail xvar set
```

will do the job. Note that any attempt to control for further variables will restrict the analysis to the comparison of cases and matched controls that share the same values of these variables. In general, this would lead to the omission of many records from the analysis. Similar considerations usually apply when investigating effect modification using the `by()` option. An important exception to this general rule is that a variable used in matching cases to controls may appear in the `by()` option without loss of data.

We illustrate the use of `mhodds` to analyze matched case–control studies using the study of endometrial cancer and exposure to oestrogens described in Breslow and Day (1980, chapter 5). In this study, there are 4 controls matched to each case, and Breslow and Day start by analyzing the 1:1 study formed by using the first control in each set. To examine the effect of exposure to oestrogen, we may use

```
. use bdendo11

. describe

Contains data from bdendo11.dta
  Obs:   126 (max= 30486)
 Vars:    13 (max=    99)
Width:    19 (max=   200)
  1. set          int    %8.0g                Set number
  2. fail         byte   %8.0g                Case=1/Control=0
  3. gall         byte   %8.0g                Gallbladder dis
  4. hyp          byte   %8.0g                Hypertension
  5. ob           byte   %8.0g                Obesity
  6. est          byte   %8.0g                Estrogen
  7. dos          byte   %8.0g                Ordinal dose
  8. dur          byte   %8.0g                Ordinal duration
  9. non          byte   %8.0g                Non-estrogen drug
 10. duration     int    %8.0g                months
 11. age          int    %8.0g                years
 12. cest         byte   %8.0g                Conjugated est dose
 13. agegrp       float  %9.0g                age group of set
Sorted by:  set

. mhodds fail est set

Mantel-Haenszel estimate of the odds ratio
Comparing est==1 vs est==0, controlling for set
WARNING: only 32 of the 63 strata formed in this analysis
contribute information about the effect of the explanatory variable

OR estimate, lower and upper 95% confidence limits, and
chi-squared test for OR=1 (1 degree of freedom)

    OR   Lower   Upper   Chisq  p_value
 9.667   2.945  31.733  21.125   0.000
```

In the case of the 1:1 matched study, the Mantel–Haenszel methods are equivalent to conditional likelihood methods. The maximum conditional likelihood estimate of the odds ratio is given by the ratio of the off-diagonal frequencies in the table

```
           | Control
      Case |         0          1 |     Total
-----------+----------------------+----------
         0 |         4          3 |         7
         1 |        29         27 |        56
-----------+----------------------+----------
      Total|        33         30 |        63
```

This is $29/3 = 9.67$, which agrees exactly with the value obtained from `mhodds`. In the more general 1:m matched study, however, the Mantel–Haenszel methods are no longer precisely the same as maximum conditional likelihood, although they usually agree quite closely.

To illustrate the use of the `by()` option in matched studies we look at the effect of exposure to oestrogen, stratified by `age3` which codes the sets (by age of case) in three groups (55–64, 65–74, and 75+), as follows:

```
. generate age3 =agegrp

. recode age3 1/2=1 3/4=2 5/6=3
(124 changes made)

. mhodds fail est set, by(age3)

Mantel-Haenszel estimate of the odds ratio
Comparing est==1 vs est==0, controlling for set
by age3
WARNING: only 32 of the 63 strata formed in this analysis
contribute information about the effect of the explanatory variable

OR estimate, lower and upper 95% confidence limits, and
chi-squared test for OR=1 (1 degree of freedom)

     age3      OR   Lower   Upper   Chisq  p_value
        1   6.000   0.722  49.837   3.571   0.059
        2  15.000   1.981  113.556  12.250   0.000
        3   8.000   1.001  63.963   5.444   0.020

Mantel-Haenszel estimate controlling for: set age3

    OR   Lower   Upper   Chisq  p_value
 9.667   2.945  31.733  21.125   0.000

Approx chisq for unequal ORs (effect modification)    0.41 (2 df, p = 0.813)
```

Note that there is no further loss of information when we stratify by `age3` because age was one of the matching variables. The full set of matched controls can be used in the same way. For example, the effect of exposure to oestrogen is obtained (using the full data set) by

```
. use bdendo

. describe

Contains data from bdendo.dta
  Obs:   315 (max= 30486)
 Vars:    13 (max=    99)
Width:    19 (max=   200)
  1. set         int    %8.0g                Set number
  2. fail        byte   %8.0g                Case=1/Control=0
  3. gall        byte   %8.0g                Gallbladder dis
  4. hyp         byte   %8.0g                Hypertension
  5. ob          byte   %8.0g                Obesity
  6. est         byte   %8.0g                Estrogen
  7. dos         byte   %8.0g                Ordinal dose
  8. dur         byte   %8.0g                Ordinal duration
  9. non         byte   %8.0g                Non-estrogen drug
 10. duration    int    %8.0g                months
 11. age         int    %8.0g                years
 12. cest        byte   %8.0g                Conjugated est dose
 13. agegrp      float  %9.0g                age group of set
Sorted by:  set

. mhodds fail est set

Mantel-Haenszel estimate of the odds ratio
Comparing est==1 vs est==0, controlling for set
WARNING: only 58 of the 63 strata formed in this analysis
contribute information about the effect of the explanatory variable

OR estimate, lower and upper 95% confidence limits, and
chi-squared test for OR=1 (1 degree of freedom)

    OR   Lower   Upper   Chisq  p_value
 8.462   3.438  20.827  31.156   0.000
```

The effect of exposure to oestrogen, stratified by `age3`, is obtained by

```
. generate age3 =agegrp

. recode age3 1/2=1 3/4=2 5/6=3
(310 changes made)

. mhodds fail est set, by(age3)

Mantel-Haenszel estimate of the odds ratio
Comparing est==1 vs est==0, controlling for set
by age3
WARNING: only 58 of the 63 strata formed in this analysis
contribute information about the effect of the explanatory variable

OR estimate, lower and upper 95% confidence limits, and
chi-squared test for OR=1 (1 degree of freedom)

     age3      OR   Lower   Upper   Chisq  p_value
        1   3.800   0.822  17.574   3.379   0.066
        2  10.667   2.788  40.814  18.689   0.000
        3  13.500   1.598  114.026   9.766   0.002

Mantel-Haenszel estimate controlling for: set age3

    OR   Lower   Upper   Chisq  p_value
 8.462   3.438  20.827  31.156   0.000

Approx chisq for unequal ORs (effect modification)    1.41 (2 df, p = 0.494)
```

References

Breslow, N. B. and N. B. Day. 1980. *Statistical Methods in Cancer Research. I: The analysis of case–control studies*. IARC Scientific Publications Number 32. Lyon, France: International Agency for Research on Cancer.

Clayton, D. G. and M. Hills. 1993. *Statistical Models in Epidemiology*. Oxford: Oxford University Press.

Plummer, M. 1995. On the variance of the Mantel–Haenszel estimator. unpublished manuscript, submitted to *Biometrics*, *Shorter Communications*.

STB categories and insert codes

Inserts in the STB are presently categorized as follows:

General Categories:			
an	announcements	*ip*	instruction on programming
cc	communications & letters	*os*	operating system, hardware, & interprogram communication
dm	data management		
dt	data sets	*qs*	questions and suggestions
gr	graphics	*tt*	teaching
in	instruction	*zz*	not elsewhere classified
Statistical Categories:			
sbe	biostatistics & epidemiology	*srd*	robust methods & statistical diagnostics
sed	exploratory data analysis	*ssa*	survival analysis
sg	general statistics	*ssi*	simulation & random numbers
smv	multivariate analysis	*sss*	social science & psychometrics
snp	nonparametric methods	*sts*	time-series, econometrics
sqc	quality control	*sxd*	experimental design
sqv	analysis of qualitative variables	*szz*	not elsewhere classified

In addition, we have granted one other prefix, *crc*, to the manufacturers of Stata for their exclusive use.

International Stata Distributors

International Stata users may also order subscriptions to the *Stata Technical Bulletin* from our International Stata Distributors.

Company: Dittrich & Partner Consulting
Address: Prinzenstrasse 2
D-42697 Solingen
Germany
Phone: +49 212-3390 99
Fax: +49 212-3390 90
Countries served: Austria, Germany

Company: Howching
Address: 11th Fl. 356 Fu-Shin N. Road
Taipei, Taiwan, R.O.C.
Phone: +886-2-505-0525
Fax: +886-2-503-1680
Countries served: Taiwan

Company: Metrika Consulting
Address: Ruddammsvagen 21
11421 Stockholm
Sweden
Phone: +46-708-163128
Fax: +46-8-6122383
Countries served: Baltic States, Denmark, Finland, Iceland, Norway, Sweden

Company: Oasis Systems BV
Address: Lekstraat 4
3433 ZB Nieuwegein
The Netherlands
Phone: +31 30 6066336
Fax: +31 30 6065844
Countries served: The Netherlands

Company: Ritme Informatique
Address: 34 boulevard Haussmann
75009 Paris, France
Phone: +33 1 42 46 00 42
Fax: +33 1 42 46 00 33
Countries served: Belgium, France, Luxembourg, Switzerland

Company: Timberlake Consultants
Address: 47 Hartfield Crescent
West Wickham
Kent BR4 9DW, U.K
Phone: +44 181 462 0495
Fax: +44 181 462 0493
Countries served: Eire, Portugal, U.K.