Bounds for STATA: Draft Version 1.0

Arie Beresteanu and Charles F. Manski
Department of Economics, Northwestern University
June 21 2000

1. Uses of the Package

The STATA routines bundled in this package implement many of the methods for nonparametric analysis of treatment response developed in Manski (1990, 1994, 1995, 1997), and Manski and Pepper (2000). The most basic of these methods yields sharp bounds on average treatment effects and other quantities of interest in the absence of maintained structural assumptions. Tighter bounds are obtained when various weak structural assumptions are maintained. This version of the package implements the bounds that hold under instrumental variable and monotone instrumental variable assumptions as well as those that hold under monotone and concave-monotone treatment response assumptions. The package also generates nonparametric point estimates of treatment effects under the assumption that treatment selection is exogenous.

A further use of the package is to perform nonparametric analysis of regressions with missing outcome data or jointly missing outcome and covariate data, implementing the methods of Manski (1989) and Horowitz and Manski (1998, 2000). These methods yield sharp bounds on regressions in the absence of assumptions on the nature of the missing data.

This documentation assumes that the reader is familiar with the structure of STATA. STATA syntax and notational conventions are maintained throughout.

2. Structure of the Package

STATA is a command driven statistical package. *Bounds* for STATA is a collection of STATA commands. In what follows, the names of all commands are in *italics*.

Each *Bounds* command has two versions. The "point" version of a *Bounds* command performs an analysis at a single covariate value of interest specified by the user and outputs results as STATA "return" variables with reserved names. Point commands are useful to researchers who wish to use *Bounds* commands as elements within STATA programs of their own device. To obtain bootstrap confidence intervals for estimates generated by point commands, the user calls the STATA bootstrap command *bs*. The names of all point commands end in the numeral "2."

The "set" version of a *Bounds* command performs an analysis at multiple covariate values of interest specified by the user and outputs results as new variables with reserved names, not as STATA return variables. The set commands internally generate optional bootstrap confidence intervals for estimates.

Bounds has no limitation on the number of observations in a dataset. However, the dimension of the covariate vector can be no larger than 4.

Elementary Routines

The elementary operations used by *Bounds* are nonparametric estimation of regressions, calculation of Silverman's rule of thumb bandwidth, and a procedure that assists the user in creating rectangular grids of covariate values to be used as the user-specified covariate values of interest in set commands. The Elementary Routines are commands performing these operations. They are

kernreg, kern2 –performs kernel estimation of regressions

silverman – computes Silverman's "rule of thumb" bandwidth for use in *kernreg* and *kern2*.

gridgen - a routine creating grids of covariate values.

Core Commands

Version 1.0 of *Bounds* contains these core commands for analysis of treatment response and for regression analysis with missing data. In what follows, the names of set commands are given first, followed by those of the corresponding point commands:

treat, treat2 – estimates "worst-case" bounds on average treatment effects; that is, bounds imposing no structural assumptions

iv, iv2 – estimates bounds on average treatment effects with a specified subset of the covariates used as instrumental variables

miv, miv2 – estimates bounds on average treatment effects with a specified covariate used as a monotone instrumental variable. (under construction)

monotone, mono2 – estimates bounds on average treatment effects when treatment response is assumed to be monotone, concave-increasing, or convex-decreasing.

exogenous, exog2 – estimates average treatment effects under the assumption that treatment selection is exogenous

outcen, outcen2 – estimates worst-case bounds on regressions when some observations have missing outcome data

jointcen, joint2 – estimates worst-case bounds on regressions when some observations have jointly missing outcome and covariate data.

The core commands share a common format, this being

command(outcome data, treatment data, covariate data, instruments, covariate values of interest, command-specific parameters).

The names of the input arguments are the same for all commands, but some commands only require a subset of the arguments. The outputs of set commands are new variables and those of point commands are STATA return variables. The new variables and return variables containing the outputs have reserved names. These names are the same across the different commands, but some commands generate only a subset of the outputs.

Elementary routine *kernreg* has the same format as the core commands except that the new variables do not have reserved names. Instead, the user defines names for these variables through the STATA Generate parameter. Routine *kern2* has the same format as the other point commands.

3. Installation of the Package

The *Bounds* software is available as a zip file on these webpages:

Charles Manski: http://www.econ.faculty.northwestern.edu/faculty/manski/

Arie Beresteanu: http://pubweb.northwestern.edu/~abe267/bounds

The file name is: bounds_stata.zip

Unzip the file to a dedicated folder. For example, one might create the folder c:\stata\bounds. These file names will appear in the folder: *kernreg.ado, kern2.ado silverman.ado, gridgen.ado, treatmen.ado, treat2.ado iv.ado, iv2.ado, monotone.ado, mono2.ado exogenous.ado, exog2.ado, outcen.ado, outcen2.ado jointcen.ado, joint2.ado,*

To use the software, you must "tell" STATA the name of the folder within which the files are stored. Write the following command in the STATA command window:

adopath + c:\stata\bounds

(replace c:\stata\bounds by your chosen folder name). STATA will then add c:\stata\bounds to its active directory. To avoid entering this command every time you open a STATA session, you can add the folder name permanently to the active path of STATA. To do this, add the above command to your profile.do program. For more details, please refer to the STATA manual description of the *adopath* command.

This completes the installation.

4. Elementary Routines

Kernel estimation of mean regressions

Purpose: use the Nadaraya-Watson kernel smoothing method to estimate the conditional expectation E(y|x). Here y is a scalar outcome variable and x is a covariate vector of dimension 4 or less. See Hardle (1990) for the theory of kernel estimation.

Syntax:

kernreg varlist(min=2 max=5) [if] [in] , Generate(string) AT(string) [W1(real 0.0) W2(real 0.0) W3(real 0.0) W4(real 0.0) BIweight EPan GAUss RECtangle TRIangle]

kern2 varlist(min=2 max=5) [if] [in] , Generate(string) AT(string) [W1(real 0.0) W2(real 0.0) W3(real 0.0) W4(real 0.0) BIweight EPan GAUss RECtangle TRIangle]

varlist:

The first variable in the varlist contains the outcome data. The rest of the variables (up to 4) contain the covariate data.

options:

Generate(string) - contains the name of the new variable to be generated as a result of this command. The name of the variable should not be of an existing variable.

AT(string) – For *kernreg*, the "string" should contain the names of up to 4 existing variables containing the covariate values of interest; that is, the values at which conditional expectations are to be estimated. For *kern2*, the "string" contains up to 4 numbers specifying the single covariate value of interest. In both cases the names or values should be separated by blanks. See the examples below for further details.

W1, W2, W3, W4 – bandwidths for up to 4 covariate components. Bandwidth specification is optional. If one or more is not specified, Silverman's rule of thumb is used.

BI, EP, GAU, REC, TRI – the kernel used (see definitions below). Kernel specification is optional. If none is specified, EP is used.

Method: For a specified covariate value x_0 and each of the n observations in the data, the routine calculates the bandwidth-normalized Euclidean distance $z_j = \sqrt{\sum_{l=1}^{k}\left(\frac{x_{0l} - x(j)_l}{h_l}\right)^2}$ where *j* goes from 1 to *n* and *l* stands for the l^{th} element of the vector x**.** The bandwidths h may be specified by the user through the W1, W2, W3, and W4 options. If no bandwidth is specified for some component of x, Silverman's rule of thumb bandwidth is used for this component.

Using the Nadaraya-Watson method, the conditional expectation is calculated as follows: $E[y \mid x = x0)] = \frac{\sum_{j=1}^{n} y(j) \cdot k(z_j)}{\sum_{j=1}^{n} k(z_j)}$.

Here $k(\cdot)$ is the kernel function. The user may specify these kernels:

EP - The Epanechnikiv density. $k(u) = \frac{3}{4}(1-u^2)$ for $|u|<1$ and zero otherwise.

GAU - The Gaussian density. $k(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$ for all *u*.

TRI - The triangular density. $k(u) = |1-u|$ for $|u|<1$ and zero otherwise.

REC - The uniform density. $k(u) = \frac{1}{2}$ for $|u|<1$ and zero otherwise.

QUA - The quartic density. $k(u) = \frac{15}{16}(1-u^2)^2$ for $|u|<1$ and zero otherwise.

The Epanechnikov kernel is used if the user does not specify a kernel or enters an invalid kernel type.

The product of the routine is the new variable specified in Generate (string).

Example: *kernreg* y x1 x2, g(yhat) at(x1value x2value) w1(2.3) w2(1.8) gau.

Here y contains the outcome variable and (x1 x2) are the covariates. The bandwidths applied to the first and second covariate components are 2.3 and 1.8. The Gaussian kernel is used.

The strings x1value and x2value are the names of existing user-specified variables that contain the covariate values of interest**.** Thus x1value contains the values of x1 and x2value contains those of x2. The two variables x1value and x2 value must have the same length, but otherwise may be specified as the user wishes. These variables can be created in two different ways. The user may manually enter the values in STATA's edit window or may use the *gridgen* procedure in this package, described later.

Silverman's rule-of-thumb bandwidth

Purpose: This routine computes Silverman's rule-of-thumb bandwidth for users who prefer to employ automated bandwidths rather than to select their own. See Silverman (1985) and Hardle (1990). The bandwidth is computed separately for each component of the covariate vector.

Syntax: *silverman* varlist(min=1)
varlist contains the name of the variables for which Silverman's rule of thumb bandwidth will be calculated.

Method: The following formula describes the computation of the bandwidth: Let

$$std = std(X_i)_{i=1..k}$$
$$q25 = (25^{th}\ quantile\ of\ X_i)_{i=1..k}$$
$$q75 = (75^{th}\ quantile\ of\ X_i)_{i=1..k}$$

Then the bandwidth vector is

$$bandwidth_i = 0.9 * \min(std_i, \frac{q75 - q25_i}{1.349}) * n^{-0.2} \qquad \text{for i = 1..k}$$

The results are output both to the screen and to the return list for each variable in the varlist.

Grid generation procedure

Purpose: This routine creates vector variables that take on all values in a rectangular grid specified by the user. This provides a simple way of generating a grid of covariate values (x1value x2value ……) for use as the argument of the AT parameter in *Bounds* set commands.

Syntax:

gridgen newvarlist(min=1 max=4), START(string) FINISH(string) JUMP(string)

newvarlist - names of up to 4 new variables to be created.

START - starting values for the variables specified in newvarlist

FINISH - ending values for the variables specified in newvarlist

JUMP – increments for the variables specified in newvarlist

Example: *gridgen* x1value x2value, start(0 2) finish(1 10) jump(0.1 1)

Two new variables are created, named x1value and x2value. Variable x1value will take values between 0 and 1 with increments of 0.1, thus 11 different values in all. Variable x2value will takes values between 2 and 10, with increments of 1, thus 9 different values in all. Hence variables x1value and x2value will each have length 11 x 9 = 99. For example, the first 9 elements of (x1value, x2value) are (0, 2), (0, 3), . . . , (0, 10), and the next 9 elements are (0.1, 2), (0.1, 3), . . . , (0.1, 10),

Note: Due to a structural characteristic of variable storage in STATA, the length of the variables x1value and x2value created by *gridgen* must not exceed the length of the data set on which estimation is to be performed.

5. Core Commands for Analysis of Treatment Response

The framework/notation for the analysis of treatment response used in this section follows Manski (1995, 1997). There is a population J of persons. Each member j of population J has observable covariates $x_j \in X$ and a response function $y_j(\cdot): T \to Y$ mapping the mutually exclusive and exhaustive treatments $t \in T$ into outcomes $y_j(t) \in Y$. Person j has a realized treatment $z_j \in T$ and a realized outcome $y_j \equiv y_j(z_j)$, both of which are observable. The latent outcomes $y_j(t)$, $t \neq z_j$ are not observable.

An empirical researcher learns the distribution P(x, z, y) of covariates, realized treatments, and realized outcomes by observing a random sample of the population. The researcher's problem is to combine this empirical evidence with assumptions in order to learn about the distribution of response functions within the sub-population defined by a specified value of the covariates x. In particular, the researcher may want to learn an average treatment effect of the form E[y(t)| x] – E[y(s)|x], where s and t are treatments.

With one exception, all of the core commands for the analysis of treatment response assume that there are two treatments, coded as 0 and 1. The exception is command *monotone*, which permits treatments to be real-valued.

Worst-Case Bounds

Purpose: This command estimate the worst-case bounds introduced in Manski (1990). The outcome variable y is assumed bounded. Before using this command, one should scale y so that its lower bound equals zero and upper bound equals one.

Syntax:
treat varlist(min=2 max=6) [if] , AT(string) [CONtinuous W1(real 0.0) W2(real 0.0) W3(real 0.0) W4(real 0.0) BIweight EPan GAUss RECtangle TRIangle Boot(integer 0)]

treat2 varlist(min=2 max=6) [if] , AT(string) [CONtinuous W1(real 0.0) W2(real 0.0) W3(real 0.0) W4(real 0.0) BIweight EPan GAUss RECtangle TRIangle]

varlist:

The first variable in varlist contains the outcome data (rescaled to be between 0 and 1). The second variable contains the binary treatment variable (0 or 1). The remaining variables (up to 4) contain the covariate data.

options:

AT(string) – See *kernreg* command.

CON – See *kernreg* command.

W1, W2,W3,W4 - See *kernreg* command.

BI, EP, GAU, REC, TRI – See *kernreg* command.

Boot – Number of bootstrap iterations, optional. Applicable for *treat* only.

Method: For each treatment t, the worst-case bounds on E[y(1)|x] and E[y(0)|x] are

$$\begin{aligned} & E[y(1) \mid x, z=1]\Pr(z=1 \mid x) \\ & \leq E[y(1) \mid x] \leq \\ & E[y(1) \mid x, z=1]\Pr(z=1 \mid x)+\Pr(z=0 \mid x) \end{aligned}$$

and

$$\begin{aligned} & E[y(0) \mid x, z=0]\Pr(z=0 \mid x) \\ & \leq E[y(0) \mid x] \leq \\ & E[y(0) \mid x, z=0]\Pr(z=0 \mid x)+\Pr(z=1 \mid x) \end{aligned}$$

The resulting bound on the average treatment effect is

$$\begin{aligned} & E[y(1) \mid x, z=1]\Pr(z=1 \mid x)-E[y(0) \mid x, z=0]\Pr(z=0 \mid x)+\Pr(z=1 \mid x) \\ & \leq E[y(1) \mid x]-E[y(0) \mid x] \leq \\ & E[y(1) \mid x, z=1]\Pr(z=1 \mid x)+\Pr(z=0 \mid x)-E[y(0) \mid x, z=0]\Pr(z=0 \mid x) \end{aligned}$$

The command uses command *kernreg* to estimate these bounds if CON is specified, and uses cell means otherwise. In the former case, the user may input the bandwidth and kernel to be used. If no values are input, the *kernreg* defaults are employed. The command *treat* outputs 10 new variables with reserved names and the command *treat2* outputs STATA return variables with the same names. The reserved names are as follows (in each case, the output is the estimate of the quantity specified):

- *prop0* = $\Pr(z = 0 \mid x = x_0)$
- *prop1* = $\Pr(z = 1 \mid x = x_0)$
- *yhat0* = $E[y(0) \mid x = x_0, z = 0]$
- *yhat1* = $E[y(1) \mid x = x_0, z = 1]$
- *boundL0* = lower bound on E[y(0)|x]
- *boundU0* =upper bound on E[y(0)|x]
- *boundL1* = lower bound on E[y(1)|x]
- *boundU1* = upper bound on E[y(1)|x]
- *treatL* = lower bound on treatment effect
- *treatU* = upper bound on treatment effect

Warning: If variables with these names have previously been specified, *treat* overwrites them.

Examples:

treat y z x1 x2, at(x1value x2value) con w2(2.6) gau

The output are 10 new variables as defined above, calculated at each of the covariate values specified in (x1value, x2value). The conditional expectations are calculated using *kernreg,* The default Silverman bandwidth is used for x1. The bandwidth for x2 is specified to be 2.6. The Gaussian kernel is used.

treat2 y z x, at(12)

The output are the 10 STATA return variables as defined above, calculated at the single covariate value x = 12. The conditional expectations are calculated using cell means.

treat2 y z x , at(12) con

The same as above except that, with "con" specified, *kern2* is used to calculate the conditional expectations. The default bandwidth and kernel are used.

Bootstrap confidence intervals:

The STATA command *bs* can be used to obtain bootstrap confidence intervals for the STATA return variables output by the point command *treat2*. Here is an example.

```
use C:\stata\data\input.dta
treat2 y z x , at(12) con w1(2.8)
bs "treat2 y z x , at(12) con w1(2.8)" "r(treatL) r(treatU)" , reps(200)
```

The first command uploads a file named input.dta from C:\stata\data. Then worst-case bounds on the average treatment effect are estimated at the covariate value x = 12 using kernel regression with bandwidth 2.8 and the Epanechnikov kernel. Then the *bs* command runs 200 bootstrap iterations to calculate 95% confidence intervals for the specified STATA return variables treatL and treatU.

The STATA command *bs* cannot be used in conjunction with the set command *treat*. Hence *treat* incorporates its own percentile bootstrap procedure to internally compute 95% bootstrap confidence intervals. To enable the internal procedure, one specifies the Boot option in the *treat* command.

When Boot is specified, *treat* generates 30 new variables with reserved names. Of these, 10 variables are the variables defined earlier: *prop0, prop1, yhat0 , yhat1, boundL0, boundU0, boundL1, boundU1, treatL, treatU* . The remaining 20 variables give the lower and upper endpoints of the 95% confidence intervals for these ten variables. The reserved names for the corresponding confidence interval endpoints are (*pr0_lb, pr0_ub), (pr1_lb, pr1_ub), (yh0_lb, yh0_ub), (yh1_lb, yh1_ub), (bdL0_lb, bdL0_ub), (bdU0_lb, bdU1_ub), (bdL1_lb, bdL1_ub), (bdU1_lb), bdU1_ub), (trL_lb, trL_ub), (trU_lb, trU_ub).*

Example: *treat* y z x1 x2 x3, at(xg1 xg2 xg3) con gau W1(2.55) w3(1.7) boot(150)

Command *treat* is run and 150 bootstrap iterations are performed to yield 95% confidence intervals for the resulting estimates.

Note: In the present version of *treat* and other *Bounds* set commands, a new set of bootstrap pseudo-samples is generated to produce confidence intervals at each covariate value specified in AT. It would be preferable to use the same pseudo-samples throughout, but the structure of STATA makes this difficult to achieve.

Estimates Assuming Exogenous Treatment Selection

Purpose: This command estimates average treatment effect under the assumption of exogenous treatment selection.

Syntax:
exogen varlist(min=2 max=6) [if] , AT(string) [CONtinuous W1(real 0.0) W2(real 0.0) W3(real 0.0) W4(real 0.0) BIweight EPan GAUss RECtangle TRIangle Boot(integer 0)]

exogen2 varlist(min=2 max=6) [if] , AT(string) [CONtinuous W1(real 0.0) W2(real 0.0) W3(real 0.0) W4(real 0.0) BIweight EPan GAUss RECtangle TRIangle]

varlist:
The first variable in the varlist contains the outcome data (rescaled to be between 0 and 1). The second variable contains the binary treatment variable (0 or 1). The remaining variables (up to 4) contain the covariate data.
options:
AT(string) – See *kernreg* command.
CON – See *kernreg* command.
W1, W2,W3,W4 - See *kernreg* command..
BI, EP, GAU, REC, TRI – See *kernreg* command.
Boot – Applicable for *exogen* only. See *treat* command.

Method: Under the exogenous selection assumption, the following holds for each treatment t:

$$E[y(1) \mid x = x0, z = 0] = E[y(1) \mid x = x0, z = 1]$$
$$E[y(0) \mid x = x0, z = 0] = E[y(0) \mid x = x0, z = 1]$$

The assumption, which is not testable, implies that

$$E[y(1) \mid x = x0, z = 0] - E[y(0) \mid x = x0, z = 1] = E(y \mid x = x0, z = 1) - E(y \mid x = x0, z = 0)$$

Thus the treatment effect is identified.

The command uses elementary routine *kernreg* to estimate these bounds if CON is specified, and uses cell means otherwise. In the former case, the user may input the bandwidth and kernel to be used. If no values are input, the *kernreg* defaults are used. The command *exogen* outputs 5 new variables with reserved names and the command *exogen2* outputs STATA return variables with the same names. The reserved names are (in each case, the output is the estimate of the quantity specified):

- *prop0* = $\Pr(z = 0 \mid x = x_0)$
- *prop1* = $\Pr(z = 1 \mid x = x_0)$
- *yhat0* = $E[y(0) \mid x = x_0, z = 0]$
- *yhat1* = $E[y(1) \mid x = x_0, z = 1]$
- *treat* = $E[y(1) \mid x = x_0, z = 1] - E[y(1) \mid x = x_0, z = 0]$

Examples: *exogen2* y z x, at(12)

exogen y z x1 x2, at(xg1 xg2) con w1(2.3) w2(1.8) tri

See the section on the *treat* and *treat2* commands for explanation of the options and the construction of bootstrap confidence intervals. Using the Boot option will generate the following 10 additional variables: (*pr0_lb pr0_ub), (pr1_lb pr1_ub), (yh0_lb yh0_ub), (yh1_lb yh1_ub), (trt_lb trt_ub)*

Instrumental Variable (IV) Bounds

Purpose: This command estimates the instrumental variable bound developed in Manski (1990, 1994). The outcome variable y is assumed bounded. Before using this command, one should scale y so that its lower bound equals zero and upper bound equals one.

Syntax:
iv varlist(min=2 max=5) [if] , Inst(string) AT(string) Niv(integer 20) [CONtinuous W1(real 0.0) W2(real 0.0) W3(real 0.0) W4(real 0.0) BIweight EPan GAUss RECtangle TRIangle Boot(integer 0)]

iv2 varlist(min=2 max=5) [if] , Inst(string) AT(string) Niv(integer 20) [CONtinuous W1(real 0.0) W2(real 0.0) W3(real 0.0) W4(real 0.0) BIweight EPan GAUss RECtangle TRIangle]

varlist:

The first variable in the varlist contains the outcome data (rescaled to be between 0 and 1). The second variable contains the binary treatment variable (0 or 1). The remaining variables (up to 4) contain the covariate data.

options:

Inst(string) – The name of the instrumental variable (one only).

AT(string) – See *kernreg* command.

Niv(integer) – If this parameter is included, the sup and inf operations described below in the description of the "Method" are taken over a grid of equally spaced values of the instrumental variable. The first value in the grid is the minimal value of the instrumental variable occurring in the dataset and the last is the maximal value. The user specifies the number of values as the integer argument of Niv.

IVG(string) – If this parameter is included, the sup and inf operations described in "Method" are taken over a set of user-specified values of the instrumental variable. The user specifies these values in the variable whose name is the argument of IVG. The user must include either the Niv or the IVG parameter, but not both.

CON – See *kernreg* command.

W1, W2, W3,W4 - See *kernreg* command.

BI, EP, GAU, REC, TRI – See *kernreg* command.

Boot –Applicable for *iv* only. See *treat* command.

(Note: In the *iv* procedure, the user can specify a different bandwidth for each covariate and each instrumental variable - see example below.)

Method: The IV bound on E[y(1)|w] is

$$\sup_{v}\{E(y \mid w = w0, V = v, z = 1)\Pr(z = 1 \mid w = w0, V = v)\}$$
$$\leq E[y(1) \mid w] \leq$$
$$\inf_{v}\{E(y \mid w = w0, V = v, z = 1)\Pr(z = 1 \mid w = w0, V = v) + \Pr(z = 0 \mid w = w0, V = v)\}$$

The IV bound on E[y(0)|w] is defined analogously. The lower (upper) bound on the treatment effect is the lower (upper) bound on E[y(1)|w] minus the upper (lower) bound on E[y(0)|w].

The *iv* command performs the above sup and inf operations over the finitely many values of the instrumental variable specified by the user through the Niv or the IVG parameter. The name of the variable used as the argument to IVG can be identical to the name of the variable used as the argument to Inst. In this case, the sup and inf are taken over all sample values of the instrumental variable.

The command uses elementary routine *kernreg* to estimate these bounds if CON is specified, and uses cell means otherwise. In the former case, the user may input the bandwidth and kernel to be used. If no values are input, the defaults are employed (see the section on *kernreg*). The command *iv* outputs 5 new variables with reserved names and the command *iv2* outputs STATA return variables with the same names. The reserved names are (in each case, the output is the estimate of the quantity specified):

- *LB0* = lower bound on E[y(0)|w = w_0]
- *UB0* = upper bound on E[y(0)|w = w_0]
- *LB1* = lower bound on E[y(1)|w = w_0]
- *UB1* = upper bound on E[y(1)|w = w_0]
- *treatL* - lower bound on E[y(1)|w = w_0] – E[y(0)|w = w_0].

- *treatU* - upper bound on E[y(1)|w = w_0] – E[y(0)|w = w_0].

Examples:

iv2 y z x, inst(v) at(10) niv(20) con w1(3.1) gau

iv y z x1 x2, inst(v) at(xg1 xg2) niv(25) w2(2.65) W3(1.7) tri

In the last example, W3(1.7) will be used to determine the bandwidth used for the instrumental variable v in the regressions in which it serves as an explanatory variable.

See the section on the *treat* and *treat2* commands for explanation of the options and the construction of bootstrap confidence intervals. Using the Boot option will generate the following 10 additional variables: (*LB0_lb, LB0_ub), (LB1_lb, LB1_ub), (UB0_lb, UB0_ub), (UB1_lb, UB1_ub), (trtL_lb, trtL_ub), (trtU_lb, trtU_ub)*

Monotone Treatment Response Bounds

Purpose: This command estimates the monotone and concave-increasing treatment response bounds developed in Manski (1997). The outcome variable need not be bounded. If only monotonicity is assumed, the treatment may be real-valued. If response is assumed to be increasing-concave, the treatment is assumed to be bounded from below, with t = 0 being the smallest possible value.

The current version of the command estimates the bound on E[y(t)|x] for a specified treatment t. Under construction is an extension of the command to estimate bounds on treatment effects E[y(t)|x] - E[y(s)|x] for specified treatments t and s.

Syntax:

monotone varlist(min=2 max=6) [if] , AT(string) [CONtinuous DECrease CONVex CONCave Low(real 0.0) High(real 1.0) W1(real 0.0) W2(real 0.0) W3(real 0.0) W4(real 0.0) BIweight EPan GAUss RECtangle TRIangle Boot(integer 0)]

mono2 varlist(min=2 max=6) [if] , AT(string) [CONtinuous DECrease CONVex CONCave Low(real 0.0) High(real 1.0) W1(real 0.0) W2(real 0.0) W3(real 0.0) W4(real 0.0) BIweight EPan GAUss RECtangle TRIangle]

varlist:

The first variable in the varlist contains the outcome data. The second variable contains the treatment variable. The remaining variables (up to 4) contain the covariate data.
options:
AT(string) – Specifies both the conjectured treatment and the values of covariates to condition on. The length of the string here is one larger than the number of covariates.
CON – See command *kernreg*.
DEC – Include this parameter if response is assumed to be decreasing in the treatment value. The default assumption is that response is increasing
CONV – Include this parameter if response is assumed increasing-concave (if the DEC parameter is included, CONV is ignored).
W1, W2, W3, W4 - See command *kernreg*.
BI, EP, GAU, REC, TRI – See command *kernreg*.
Boot – Applicable for *monotone* only. See command *treat*.

Method: Consider the case in which the response is assumed increasing in the treatment. The bound on $E[y(t)|x = x_0]$ is

$$K_0 P(t < z) + E(y \mid t \geq z) \cdot P(t \geq z) \leq E[y(t)] \leq K_1 P(t > z) + E(y \mid t \leq z) \cdot P(t \leq z)$$

If the response is assumed to be concave-increasing in the treatment, the bound is

$$E(yt/z \mid t < z) \cdot P(t < z) + E(y \mid t \geq z) P(t \geq z) \leq E[y(t)]$$
$$\leq E(yt/z \mid t > z) \cdot P(t > z) + E(y \mid t \leq z) \cdot P(t \leq z)$$

The command uses elementary routine *kernreg* to estimate these bounds if CON is specified, and uses cell means otherwise. In the former case, the user may input the bandwidth and kernel to be used. If no values are input, the defaults are employed (see the section on *kernreg*). The command *monotone* outputs 4 new variables with reserved names and the command *mono2* outputs STATA return variables with the same names. The reserved names are (in each case, the output is the estimate of the quantity specified):

- *prop0* = The estimate for Pr(z < t).

- *prop1* = The estimate for Pr(z ≥ t).
- *LB* = The estimate for the lower bound of E[y(t)|x = x_0].
- *UB* = The estimate for the upper bound of E[y(t)|x = x_0].

Examples:

mono2 y z x, at(3 12) con

monotone y z x1 x2, at (zgrid x1value x2value) con gau w1(3)

See the section on the *treat* and *treat2* commands for explanation of the options and the construction of bootstrap confidence intervals. Using the Boot option will generate the following 8 additional variables: (*pr0_lb, pr0_ub), (pr1_lb, pr1_ub), (LB_lb, LB_ub), (UB_lb, UB_ub)*

6. Core Commands for Analysis of Regressions with Missing Data

The framework/notation used in this section follows Manski (1989) and Horowitz and Manski (1998). There is a population J of persons. Each member j of population J has observable covariates $x_j \in X$ and a bounded outcome y_j. Outcomes should be scaled so as to have lower bound 0 and upper bound 1. The researcher wants to learn the conditional expectation E(y|x). The binary variable z_j takes the value 1 if (y_j, x_j) is observed and 0 if data are missing. The nature of the missing data differs across the commands below.

Censoring of outcome variables

Purpose: This command estimates the worst-case bound when some outcome data are missing, but covariate data are always observed. See Manski (1989).

Syntax:
outcen varlist(min=2 max=6) [if] , AT(string) [CONtinuous W1(real 0.0) W2(real 0.0) W3(real 0.0) W4(real 0.0) BIweight EPan GAUss RECtangle TRIangle Boot(integer 0)]

outcen2 varlist(min=2 max=6) [if] , AT(string) [CONtinuous W1(real 0.0) W2(real 0.0) W3(real 0.0) W4(real 0.0) BIweight EPan GAUss RECtangle TRIangle]

varlist:

The first variable in the varlist contains the outcome data (rescaled to be between 0 and 1). The second variable contains the missing data indicator (0 if data missing, 1 if present). The remaining variables (up to 4) contain the covariate data.

options:

AT(string) – See *kernreg* command.

CON – See command *kernreg*.

W1, W2,W3,W4 - See command *kernreg*.

BI, EP, GAU, REC, TRI - See command *kernreg*.

Boot – Applicable for *outcen* only. See command *treat*.

Method: The bound on E(y|x) is

$$E(y \mid x, z=1)\Pr(z=1 \mid X) \leq E(y \mid x) \leq E(y \mid x, z=1)\Pr(z=1 \mid x) + \Pr(z=0 \mid x)$$

The command *outcen* outputs 5 new variables with reserved names and the command *outcen2* outputs STATA return variables with the same names. The reserved names are (in each case, the output is the estimate of the quantity specified):

- *prop0* = Pr(z = 0| x = x_0); probability of missing data conditional on x = x_0
- *prop1* = Pr(z = 1|x = x_0); probability of observing y, conditional on x = x_0
- *yhat1* = E(y|x = x_0, z = 1), estimate of E(y|x) under assumption of exogenous nonresponse
- *boundL* = lower bound on E(y|x = x_0)
- *boundU* = upper bound on E(y|x = x_0)

Examples:

outcen y z x1 x2, at(x1value x2value) con w2(2.6) gau

outcen2 y z x, at(3)

See the section on the *treat* and *treat2* commands for explanation of the options and the construction of bootstrap confidence intervals. Using the Boot option will generate the following 10 additional variables: (*pr0_lb, pr0_ub), (pr1_lb, pr1_ub), (yh1_lb, yh1_ub), (bndL_lb, bndL_ub), (bndU_lb, bndU_ub)*

Joint Censoring of Outcomes and Covariates

Purpose: This command estimates the worst-case bound when some (outcome, covariate) data are jointly missing. See Horowitz and Manski (1998).

Syntax:

joint varlist(min=2 max=6) [if] , Low(real) High(real) [Boot(integer 0)]

joint2 varlist(min=2 max=6) [if] , Low(real) High(real)

varlist:

The first variable in the varlist contains the outcome data (rescaled to be between 0 and 1). The second variable contains the missing data indicator (0 if data missing, 1 if present). The remaining variables (up to 4) contain the covariate data.

options:

Low(integer) – lower bound of the interval of interest.

High(integer) – upper bound of the interval of interest.

Boot – Applicable for *joint* only. See command *treat*.

Method: The worst-case bound under joint censoring has the same form as the one under outcome censoring, except that P(z = 1|x) is replaced by *the effective response probability*

$$Pe(z=1 \mid x_l \le x \le x_u) = \frac{\Pr(x_l \le x \le x_u \mid z=1)\cdot \Pr(z=1)}{\Pr(x_l \le x \le x_u \mid z=1)\cdot \Pr(z=1) + \Pr(z=0)}$$

Continuous and discrete covariates are treated in same way in the *jointcen* procedure. If the covariates are discrete and the probability that x = x0 is positive, then the user may choose xl = xu = x0. If the covariates are continuous, then the user has to choose xl < xu in order to get a positive Pe. In both cases the probability $P(Y \mid x_l \le x \le x_u, z = 1)$ is calculated using cell means (and the same when z = 0).

The command *joint* outputs 3 new variables with reserved names and the command *joint2* outputs STATA return variables with the same names. The reserved names are (in each case, the output is the estimate of the quantity specified):

- *probL* = the effective response probability $P_e(z = 1|xl \le x \le xu)$
- *yhatL* = lower bound on $E(y|xl \le x \le xu)$
- *yhatU* = lower bound on $E(y|xl \le x \le xu)$

Examples:

joint y z x1 x2, at(x1value x2value) con w2(2.6) gau

joint2 y z x, at(3)

See the section on the *treat* and *treat2* commands for explanation of the options and the construction of bootstrap confidence intervals. Using the Boot option will generate the following 6 additional variables: (*prL_lb, prL_ub), (yhL_lb, yhL_ub), (yhU_lb, yhU_ub)*

References

Efron, B. and R. Tibshirani (1993), Introduction to the Bootstrap, London: Chapman & Hall.

Hardle, W. (1990), Applied Nonparametric Regression Analysis, New York: Cambridge University Press.

Horowitz, J. and C. Manski (1998), "Censoring of Outcomes and Covariates due to Survey Nonresponse: Identification and Estimation Using Weights and Imputations," Journal of Econometrics, 84, 37-58.

Horowitz, J. and C. Manski (2000), "Nonparametric Analysis of Randomized Experiments With Missing Covariate and Outcome Data," Journal of the American Statistical Association, forthcoming.

Manski, C. (1989), "Anatomy of the Selection Problem," Journal of Human Resources, 24, 343-360.

Manski, C. (1990), "Nonparametric Bounds on Treatment Effects," American Economic Review Papers and Proceedings, 80, 319-323.

Manski, C. (1994), "The Selection Problem," in C. Sims (editor), Advances in Econometrics, Sixth World Congress, Cambridge, UK: Cambridge University Press.

Manski, C. (1995), Identification Problems in the Social Sciences, Harvard University Press.

Manski, C. (1997), "Monotone Treatment Response," Econometrica, 65, 1311 - 1334.

Manski, C. and J. Pepper (2000), "Monotone Instrumental Variables: With an Application to the Returns to Schooling," Econometrica, forthcoming.

Silverman, B. (1986), Density Estimation for Statistics and Data Analysis, London: Chapman & Hall.