```
STata®
Statistics/Data Analysis
```

help binsprobit

_____

Title

   binsprobit —— Data-Driven Binscatter Probit Estimation with Robust Inference
      Procedures and Plots.

Syntax

   binsprobit _depvar_ _indvar_ [_othercovs_] [_if_] [_in_] [_weight_] [, **deriv(**_v_**)** **at(**_position_**)**
               nolink
               dots(_dotsopt_**)** **dotsgrid(**_dotsgridoption_**)** **dotsplotopt(**_dotsoption_**)**
               line(_lineopt_**)** **linegrid(**#**)** **lineplotopt(**_lineoption_**)**
               ci(_ciopt_**)** **cigrid(**_cigridoption_**)** **ciplotopt(**_rcapoption_**)**
               cb(_cbopt_**)** **cbgrid(**#**)** **cbplotopt(**_rareaoption_**)**
               polyreg(_p_**)** **polyreggrid(**#**)** **polyregcigrid(**#**)** **polyregplotopt(**_lineoption_**)**
               by(_varname_**)** **bycolors(**_colorstylelist_**)** **bysymbols(**_symbolstylelist_**)**
               bylpatterns(_linepatternstylelist_**)**
               nbins(_nbinsopt_**)** **binspos(**_position_**)** **binsmethod(**_method_**)** **nbinsrot(**#**)**
               samebinsby **randcut(**#**)**
               pselect(_numlist_**)** **sselect(**_numlist_**)**
               nsims(#**)** **simsgrid(**#**)** **simsseed(**_seed_**)**
               dfcheck(_n1 n2_**)** **masspoints(**_masspointsoption_**)**
               vce(_vcetype_**)** **asyvar(**_on/off_**)**
               level(_level_**)** **probitopt(**_probit_option_**)** **usegtools(**_on/off_**)** **noplot**
               savedata(_filename_**)** **replace**
               plotxrange(_min max_**)** **plotyrange(**_min max_**)** _twoway_options_]

   where _depvar_ is the dependent variable, _indvar_ is the independent variable for
      binning, and _othercovs_ are other covariates to be controlled for.

   The degree of the piecewise polynomial p, the number of smoothness constraints s,
      and the derivative order v are integers satisfying 0 <= s,v <= p, which can
      take different values in each case.

   fweights and **pweight**s are allowed; see weight.

Description

   binsprobit implements binscatter probit estimation with robust inference
      procedures and plots, following the results in Cattaneo, Crump, Farrell and
      Feng (2022a). Binscatter provides a flexible way to describe the mean
      relationship between two variables, after possibly adjusting for other
      covariates, based on partitioning/binning of the independent variable of
      interest. The main purpose of this command is to generate binned scatter
      plots with curve estimation with robust pointwise confidence intervals and
      uniform confidence band. If the binning scheme is not set by the user, the
      companion command binsregselect is used to implement binscatter in a
      data-driven way. Hypothesis testing for parametric specifications of and
      shape restrictions on the regression function can be conducted via the
      companion command binstest. Hypothesis testing for pairwise group comparisons
      can be conducted via the companion command binspwc. Binscatter estimation
      based on the least squares method can be conducted via the command binsreg.

   A detailed introduction to this command is given in Cattaneo, Crump, Farrell and
      Feng (2022b). Companion R and Python packages with the same capabilities are
      available (see website below).

   Companion commands: binstest for hypothesis testing of parametric specifications
      and shape restrictions, binspwc for hypothesis testing for pairwise group
      comparisons, and binsregselect for data-driven binning selection.

   Related Stata, R and Python packages are available in the following website:

      https://nppackages.github.io/

Options

```
                        Estimand
```

deriv(*v***)** specifies the derivative order of the regression function for estimation
    and plotting. The default is **deriv(0)**, which corresponds to the function
    itself.

at(*position***)** specifies the values of *othercovs* at which the estimated function is
    evaluated for plotting. The default is **at(mean)**, which corresponds to the
    mean of *othercovs*. Other options are: **at(median)** for the median of *othercovs*,
    at(0) for zeros, and **at(filename)** for particular values of *othercovs* saved in
    another file.

Note: When **at(mean)** or **at(median)** is specified, all factor variables in *othercovs*
    (if specified) are excluded from the evaluation (set as zero).

nolink specifies that the function within the inverse link (logistic) function be
    reported instead of the conditional probability function.

```
                        Dots
```

dots(*dotsopt***)** sets the degree of polynomial and the number of smoothness for point
    estimation and plotting as "dots". If **dots(p s)** is specified, a piecewise
    polynomial of degree *p* with *s* smoothness constraints is used. The default is
    dots(0 0), which corresponds to piecewise constant (canonical binscatter). If
    dots(T) is specified, the default **dots(0 0)** is used unless the degree *p* and
    smoothness *s* selection is requested via the option **pselect()** (see more details
    in the explanation of **pselect()**). If **dots(F)** is specified, the dots are not
    included in the plot.

dotsgrid(*dotsgridoption***)** specifies the number and location of dots within each bin
    to be plotted. Two options are available: *mean* and a *numeric* non-negative
    integer. The option **dotsgrid(***mean***)** adds the sample average of *indvar* within
    each bin to the grid of evaluation points. The option **dotsgrid(***#***)** adds *#*
    number of evenly-spaced points to the grid of evaluation points for each bin.
    Both options can be used simultaneously: for example, **dotsgrid(***mean 5***)**
    generates six evaluation points within each bin containing the sample mean of
    indvar within each bin and five evenly-spaced points. Given this choice, the
    dots are point estimates evaluated over the selected grid within each bin.
    The default is **dotsgrid(***mean***)**, which corresponds to one dot per bin evaluated
    at the sample average of *indvar* within each bin (canonical binscatter).

dotsplotopt(*dotsoption***)** standard graphs options to be passed on to the <u>twoway</u>
    command to modify the appearance of the plotted dots.

```
                        Line
```

line(*lineopt***)** sets the degree of polynomial and the number of smoothness
    constraints for plotting as a "line". If **line(p s)** is specified, a piecewise
    polynomial of degree *p* with *s* smoothness constraints is used. If **line(T)** is
    specified, **line(0 0)** is used unless the degree *p* and smoothness *s* selection is
    requested via the option **pselect()** (see more details in the explanation of
    pselect()). If **line(F)** or **line()** is specified, the line is not included in
    the plot. The default is **line()**.

linegrid(*#***)** specifies the number of evaluation points of an evenly-spaced grid
    within each bin used for evaluation of the point estimate set by the **line(p s)**
    option. The default is **linegrid(20)**, which corresponds to 20 evenly-spaced
    evaluation points within each bin for fitting/plotting the line.

lineplotopt(*lineoption***)** standard graphs options to be passed on to the <u>twoway</u>
    command to modify the appearance of the plotted line.

```
                        Confidence Intervals
```

ci(*ciopt***)** specifies the degree of polynomial and the number of smoothness
constraints for constructing confidence intervals. If **ci(p s)** is specified, a
piecewise polynomial of degree *p* with *s* smoothness constraints is used. If
ci(T) is specified, **ci(1 1)** is used unless the degree *p* and smoothness *s*
selection is requested via the option **pselect()** (see more details in the
explanation of **pselect()**). If **ci(F)** or **ci()** is specified, the confidence
intervals are not included in the plot. The default is **ci()**.

cigrid(*cigridoption***)** specifies the number and location of evaluation points in the
grid used to construct the confidence intervals set by the **ci(p s)** option.
Two options are available: *mean* and a *numeric* non-negative integer. The
option **cigrid(***mean***)** adds the sample average of *indvar* within each bin to the
grid of evaluation points. The option **cigrid(#)** adds # number of
evenly-spaced points to the grid of evaluation points for each bin. Both
options can be used simultaneously: for example, **cigrid(***mean 5***)** generates six
evaluation points within each bin containing the sample mean of *indvar* within
each bin and five evenly-spaced points. The default is **cigrid(***mean***)**, which
corresponds to one evaluation point set at the sample average of *indvar* within
each bin for confidence interval construction.

ciplotopt(*rcapoption***)** standard graphs options to be passed on to the <u>twoway</u>
command to modify the appearance of the confidence intervals.

Confidence Band

cb(*cbopt***)** specifies the degree of polynomial and the number of smoothness
constraints for constructing the confidence band. If **cb(p s)** is specified, a
piecewise polynomial of degree *p* with *s* smoothness constraints is used. If
the option **cb(T)** is specified, **cb(1 1)** is used unless the degree *p* and
smoothness *s* selection is requested via the option **pselect()** (see more details
in the explanation of **pselect()**). If **cb(F)** or **cb()** is specified, the
confidence band is not included in the plot. The default is **cb()**.

cbgrid(#) specifies the number of evaluation points of an evenly-spaced grid
within each bin used for evaluation of the point estimate set by the **cb(p s)**
option. The default is **cbgrid(20)**, which corresponds to 20 evenly-spaced
evaluation points within each bin for confidence band construction.

cbplotopt(*rareaoption***)** standard graphs options to be passed on to the <u>twoway</u>
command to modify the appearance of the confidence band.

Global Polynomial Regression

polyreg(*p***)** sets the degree *p* of a global polynomial regression model for plotting.
By default, this fit is not included in the plot unless explicitly specified.
Recommended specification is **polyreg(3)**, which adds a cubic polynomial fit of
the regression function of interest to the binned scatter plot.

polyreggrid(#) specifies the number of evaluation points of an evenly-spaced grid
within each bin used for evaluation of the point estimate set by the
polyreg(p) option. The default is **polyreggrid(20)**, which corresponds to 20
evenly-spaced evaluation points within each bin for confidence interval
construction.

polyregcigrid(#) specifies the number of evaluation points of an evenly-spaced
grid within each bin used for constructing confidence intervals based on
polynomial regression set by the **polyreg(p)** option. The default is
polyregcigrid(0), which corresponds to not plotting confidence intervals for
the global polynomial regression approximation.

polyregplotopt(*lineoption***)** standard graphs options to be passed on to the <u>twoway</u>
command to modify the appearance of the global polynomial regression fit.

Subgroup Analysis

by(_varname_**)** specifies the variable containing the group indicator to perform
    subgroup analysis; both numeric and string variables are supported. When
    by(_varname_**)** is specified, **binsprobit** implements estimation and inference for
    each subgroup separately, but produces a common binned scatter plot. By
    default, the binning structure is selected for each subgroup separately, but
    see the option **samebinsby** below for imposing a common binning structure across
    subgroups.

bycolors(_colorstylelist_**)** specifies an ordered list of colors for plotting each
    subgroup series defined by the option **by()**.

bysymbols(_symbolstylelist_**)** specifies an ordered list of symbols for plotting each
    subgroup series defined by the option **by()**.

bylpatterns(_linepatternstylelist_**)** specifies an ordered list of line patterns for
    plotting each subgroup series defined by the option **by()**.

    ┌─ Binning/Degree/Smoothness Selection ─┐

nbins(_nbinsopt_**)** sets the number of bins for partitioning/binning of _indvar_. If
    nbins(T) or **nbins()** (default) is specified, the number of bins is selected via
    the companion command <u>binsregselect</u> in a data-driven, optimal way whenever
    possible. If a <u>numlist</u> with more than one number is specified, the number of
    bins is selected within this list via the companion command <u>binsregselect</u>.

binspos(_position_**)** specifies the position of binning knots. The default is
    binspos(qs), which corresponds to quantile-spaced binning (canonical
    binscatter). Other options are: **es** for evenly-spaced binning, or a <u>numlist</u>
    for manual specification of the positions of inner knots (which must be within
    the range of _indvar_).

binsmethod(_method_**)** specifies the method for data-driven selection of the number of
    bins via the companion command <u>binsregselect</u>. The default is **binsmethod(dpi)**,
    which corresponds to the IMSE-optimal direct plug-in rule. The other option
    is: **rot** for rule of thumb implementation.

nbinsrot(_#_**)** specifies an initial number of bins value used to construct the DPI
    number of bins selector. If not specified, the data-driven ROT selector is
    used instead.

samebinsby forces a common partitioning/binning structure across all subgroups
    specified by the option **by()**. The knots positions are selected according to
    the option **binspos()** and using the full sample. If **nbins()** is not specified,
    then the number of bins is selected via the companion command <u>binsregselect</u>
    and using the full sample.

randcut(_#_**)** specifies the upper bound on a uniformly distributed variable used to
    draw a subsample for bins/degree/smoothness selection. Observations for which
    runiform()<=# are used. # must be between 0 and 1. By default, max(5,000,
    0.01n) observations are used if the samples size n>5,000.

pselect(_numlist_**)** specifies a list of numbers within which the degree of polynomial
    _p_ for point estimation is selected. Piecewise polynomials of the selected
    optimal degree _p_ are used to construct dots or line if **dots(T)** or **line(T)** is
    specified, whereas piecewise polynomials of degree _p+1_ are used to construct
    confidence intervals or confidence band if **ci(T)** or **cb(T)** is specified.

sselect(_numlist_**)** specifies a list of numbers within which the number of smoothness
    constraints _s_ for point estimation. Piecewise polynomials with the selected
    optimal _s_ smoothness constraints are used to construct dots or line if **dots(T)**
    or **line(T)** is specified, whereas piecewise polynomials with _s+1_ constraints
    are used to construct confidence intervals or confidence band if **ci(T)** or
    cb(T) is specified. If not specified, for each value _p_ supplied in the option
    pselect(), only the piecewise polynomial with the maximum smoothness is
    considered, i.e., _s=p_.

Note: To implement the degree or smoothness selection, in addition to **pselect()** or
    sselect(), **nbins(#)** must be specified.

        ┌─ Simulation └───

nsims(#**)** specifies the number of random draws for constructing confidence bands.
     The default is **nsims(500),** which corresponds to 500 draws from a standard
     Gaussian random vector of size [(p+1)*J − (J−1)*s]. A large number of random
     draws is recommended to obtain the final results.

simsgrid(#**)** specifies the number of evaluation points of an evenly-spaced grid
     within each bin used for evaluation of the supremum operation needed to
     construct confidence bands. The default is **simsgrid(20),** which corresponds to
     20 evenly-spaced evaluation points within each bin for approximating the
     supremum operator. A large number of evaluation points is recommended to
     obtain the final results.

simsseed(#**)** sets the seed for simulations.

        ┌─ Mass Points and Degrees of Freedom └──────────────────────

dfcheck(*n1 n2***)** sets cutoff values for minimum effective sample size checks, which
     take into account the number of unique values of *indvar* (i.e., adjusting for
     the number of mass points), number of clusters, and degrees of freedom of the
     different statistical models considered. The default is **dfcheck(20 30).** See
     Cattaneo, Crump, Farrell and Feng (2022b) for more details.

masspoints(*masspointsoption***)** specifies how mass points in *indvar* are handled. By
     default, all mass point and degrees of freedom checks are implemented.
     Available options:
     masspoints(*noadjust***)** omits mass point checks and the corresponding effective
     sample size adjustments.
     masspoints(*nolocalcheck***)** omits within-bin mass point and degrees of freedom
     checks.
     masspoints(*off***)** sets **masspoints(***noadjust***)** and **masspoints(***nolocalcheck***)**
     simultaneously.
     masspoints(*veryfew***)** forces the command to proceed as if *indvar* has only a few
     number of mass points (i.e., distinct values). In other words, forces the
     command to proceed as if the mass point and degrees of freedom checks were
     failed.

        ┌─ Standard Error └───

vce(*vcetype***)** specifies the *vcetype* for variance estimation used by the command
     probit. The default is **vce(robust).**

asyvar(*on/off***)** specifies the method used to compute standard errors. If
     asyvar(on) is specified, the standard error of the nonparametric component is
     used and the uncertainty related to other control variables *othercovs* is
     omitted. Default is **asyvar(off),** that is, the uncertainty related to
     othercovs is taken into account.

        ┌─ Other Options └──

level(#**)** sets the nominal confidence level for confidence interval and confidence
     band estimation. Default is **level(95).**

probitopt(*probit_option***)** options to be passed on to the command probit. For
     example, options that control for the optimization process can be added here.

usegtools(*on/off***)** forces the use of several commands in the community-distributed
     Stata package **gtools** to speed the computation up, if *on* is specified. Default
     is **usegtools(off).**

For more information about the package **gtools,** please see
     https://gtools.readthedocs.io/en/latest/index.html.

noplot omits binscatter plotting.

savedata(_filename_**)** specifies a filename for saving all data underlying the
binscatter plot (and more).

replace overwrites the existing file when saving the graph data.

plotxrange(_min max_**)** specifies the range of the x-axis for plotting. Observations
outside the range are dropped in the plot.

plotyrange(_min max_**)** specifies the range of the y-axis for plotting. Observations
outside the range are dropped in the plot.

_twoway_options_ any unrecognized options are appended to the end of the twoway
command generating the binned scatter plot.

Examples

Setup
    . sysuse auto

Run a binscatter probit regression and report the plot
    . binsprobit foreign weight mpg

Add confidence intervals and confidence band
    . binsprobit foreign weight mpg, ci(1 1) nbins(5)

Stored results

Scalars
e(N)	number of observations
e(level)	confidence level
e(dots_p)	degree of polynomial for dots
e(dots_s)	smoothness of polynomial for dots
e(line_p)	degree of polynomial for line
e(line_s)	smoothness of polynomial for line
e(ci_p)	degree of polynomial for confidence interval
e(ci_s)	smoothness of polynomial for confidence interval
e(cb_p)	degree of polynomial for confidence band
e(cb_s)	smoothness of polynomial for confidence band

Matrices
e(N_by)	number of observations for each group
e(Ndist_by)	number of distinct values for each group
e(Nclust_by)	number of clusters for each group
e(nbins_by)	number of bins for each group
e(cval_by)	critical value for each group, used for confidence bands
e(imse_var_rot)	variance constant in IMSE, ROT selection
e(imse_bsq_rot)	bias constant in IMSE, ROT selection
e(imse_var_dpi)	variance constant in IMSE, DPI selection
e(imse_bsq_dpi)	bias constant in IMSE, DPI selection

References

Cattaneo, M. D., R. K. Crump, M. H. Farrell, and Y. Feng. 2022a. On Binscatter.
arXiv:1902.09608.

Cattaneo, M. D., R. K. Crump, M. H. Farrell, and Y. Feng. 2022b. Binscatter
Regressions. _arXiv:1902.09615_.

Authors

Matias D. Cattaneo, Princeton University, Princeton, NJ. cattaneo@princeton.edu.

Richard K. Crump, Federal Reserve Band of New York, New York, NY.
richard.crump@ny.frb.org.

Max H. Farrell, University of Chicago, Chicago, IL. max.farrell@chicagobooth.edu.

Yingjie Feng, Tsinghua University, Beijing, China. fengyingjiepku@gmail.com.