stteffects postestimation — Postestimation tools for stteffects

Postestimation commands

The following postestimation commands are of special interest after `stteffects`:

Command	Description
`teoverlap`	overlap plots
`tebalance`	check balance of covariates

The following standard postestimation commands are also available:

Command	Description
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	propensity scores, censored survival probability, etc.
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

Description for predict

`predict` creates a new variable containing predictions such as treatment effects, conditional means, propensity scores, linear predictions, and log square roots of latent variances.

Menu for predict

Statistics > Postestimation

Syntaxes for predict

Syntaxes are presented under the following headings:

Syntax for predict after stteffects ipw

```
predict [type] {stub* | newvar | newvarlist} [if] [in]
        [ , statistic tlevel(treat_level) ]

predict [type] stub* [if] [in], scores
```

statistic	Description
Main	
`ps`	propensity score; the default
`censurv`	censored survival probability
`xb`	linear prediction for propensity score
`cxb`	linear prediction for censoring model
`lnsigma`	log square root of latent variance (for treatment model `hetprobit()`)
`clnshape`	log of conditional latent shape (for censoring distribution Weibull, log normal, or gamma)

If you do not specify `tlevel()` and only specify one new variable, `ps` assumes `tlevel()` specifies the first treatment level.

If you do not specify `tlevel()` and only specify one new variable, `xb` and `lnsigma` assume `tlevel()` specifies the first noncontrol treatment level.

You specify one or t new variables with `ps`, where t is the number of treatment levels.

You specify one or $t - 1$ new variables with `xb` and `lnsigma`.

You specify one new variable with `censurv`, `cxb`, and `clnshape`.

Syntax for predict after stteffects ipwra

```
predict [type] {stub* | newvar | newvarlist} [if] [in]
        [ , statistic tlevel(treat_level) ]

predict [type] stub* [if] [in], scores
```

statistic	Description
Main	
te	treatment effect; the default
cmean	conditional mean at treatment level
ps	propensity score
censurv	censored survival probability
xb	linear prediction for outcome model
cxb	linear prediction for censoring model
psxb	linear prediction for propensity score
lnshape	log of conditional latent shape (for outcome distribution Weibull, log normal, or gamma) at treatment level
clnshape	log of conditional latent shape (for censoring distribution Weibull, log normal, or gamma)
pslnsigma	log square root of latent variance (for treatment model hetprobit()) for propensity score

If you do not specify tlevel() and only specify one new variable, te and psxb assume tlevel() specifies the first non-control treatment level.

If you do not specify tlevel() and only specify one new variable, cmean, ps, xb, and pslnsigma assume tlevel() specifies the first treatment level.

You specify one or t new variables with cmean, ps, xb, and lnshape, where t is the number of treatment levels.

You specify one or $t-1$ new variables with te, psxb, and pslnsigma.

You specify one new variable with censurv, cxb, and clnshape.

Syntax for predict after stteffects ra

```
predict [type] {stub* | newvar | newvarlist} [if] [in]
        [ , statistic tlevel(treat_level) ]

predict [type] stub* [if] [in], scores
```

statistic	Description
Main	
te	treatment effect; the default
cmean	conditional mean at treatment level
xb	linear prediction for outcome model
lnshape	log of conditional latent shape (for outcome distribution Weibull, log normal, or gamma) at treatment level

If you do not specify tlevel() and only specify one new variable, te assumes tlevel() specifies the first noncontrol treatment level.

If you do not specify tlevel() and only specify one new variable, cmean, xb, and lnshape assume tlevel() specifies the first treatment level.

You specify one or t new variables with cmean, xb, and lnshape, where t is the number of treatment levels.

You specify one or $t - 1$ new variables with te.

Syntax for predict after stteffects wra

```
predict [type] {stub* | newvar | newvarlist} [if] [in]
        [ , statistic tlevel(treat_level) ]

predict [type] stub* [if] [in], scores
```

statistic	Description
Main	
te	treatment effect; the default
cmean	conditional mean at treatment level
censurv	censored survival probability
xb	linear prediction for outcome model
cxb	linear prediction for censoring model
lnshape	log of conditional latent shape (for outcome distribution Weibull, log normal, or gamma) at treatment level
clnshape	log of conditional latent shape (for censoring distribution Weibull, log normal, or gamma)

If you do not specify tlevel() and only specify one new variable, te assumes tlevel() specifies the first noncontrol treatment level.

If you do not specify tlevel() and only specify one new variable, cmean, xb, and lnshape assume tlevel() specifies the first treatment level.

You specify one or t new variables with cmean, xb, and lnshape, where t is the number of treatment levels.

You specify one or $t - 1$ new variables with te.

You specify one new variable with censurv, cxb, and clnshape.

Options for predict

Options are presented under the following headings:

Options for predict after stteffects ipw

Main

`ps`, the default, calculates the propensity score of each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`censurv` calculates the survivor probability from the time-to-censoring model. (In other words, it calculates the probability that an outcome is not censored.) This option is allowed only if a censoring model is specified at estimation time. You need to specify only one new variable.

`xb` calculates the propensity score linear prediction at each noncontrol level of the treatment or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

`cxb` calculates the linear prediction of the censoring model. This option is allowed only if a censoring model is specified at estimation time. You need to specify only one new variable.

`lnsigma` calculates the log square root of the latent variance. This option is valid only when treatment model `hetprobit()` is used. You need to specify only one new variable.

`clnshape` calculates the log of the conditional latent shape parameter of the censoring distribution. This option is valid when censoring distribution Weibull, log normal, or gamma is used. You need to specify only one new variable.

`tlevel(`*treat_level*`)` specifies the treatment level for prediction.

`scores` calculates the score variables. Parameter-level scores are computed for the treatment mean and average treatment-effect equations. Equation-level scores are computed for the censoring and propensity-score equations.

The jth new variable will contain the scores for the jth parameter in the coefficient table if $j \leq t$, where t is the number of treatment levels. Otherwise, it will contain the scores for fitted equation $j-t$ following the first t parameters in the coefficient table.

Options for predict after stteffects ipwra

Main

`te`, the default, calculates the treatment effect for each noncontrol treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

`cmean` calculates the conditional mean for each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

ps calculates the propensity score of each treatment level or the treatment level specified in tlevel(). If you specify the tlevel() option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

censurv calculates the survivor probability from the time-to-censoring model. (In other words, it calculates the probability that an outcome is not censored.) This option is allowed only if a censoring model is specified at estimation time. You need to specify only one new variable.

xb calculates the outcome model linear prediction at each treatment level or the treatment level specified in tlevel(). If you specify the tlevel() option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

cxb calculates the linear prediction of the censoring model. This option is allowed only if a censoring model is specified at estimation time. You need to specify only one new variable.

psxb calculates the propensity score linear prediction at each noncontrol level of the treatment or the treatment level specified in tlevel(). If you specify the tlevel() option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

lnshape calculates the log of the conditional latent shape parameter for each treatment level or the treatment level specified in tlevel(). This option is valid when outcome distribution Weibull, log normal, or gamma is used. If you specify the tlevel() option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

clnshape calculates the log of the conditional latent shape parameter for the censoring distribution. This option is valid when censoring distribution Weibull, log normal, or gamma is used. You need to specify only one new variable.

pslnsigma calculates the log square root of the latent variance for the propensity score. This option is valid only when treatment model hetprobit() is used. You need to specify only one new variable.

tlevel(*treat_level*) specifies the treatment level for prediction.

scores calculates the score variables. Parameter-level scores are computed for the treatment mean and average treatment-effect equations. Equation-level scores are computed for the outcome, censoring, and propensity-score equations.

The jth new variable will contain the scores for the jth parameter in the coefficient table if $j \leq t$, where t is the number of treatment levels. Otherwise, it will contain the scores for fitted equation $j-t$ following the first t parameters in the coefficient table.

Options for predict after stteffects ra

Main

te, the default, calculates the treatment effect for each noncontrol treatment level or the treatment level specified in tlevel(). If you specify the tlevel() option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

cmean calculates the conditional mean for each treatment level or the treatment level specified in tlevel(). If you specify the tlevel() option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`xb` calculates the outcome model linear prediction at each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`lnshape` calculates the log of the conditional latent shape parameter for each treatment level or the treatment level specified in `tlevel()`. This option is valid when the outcome distribution Weibull, log normal, or gamma is used. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`tlevel(`*treat_level*`)` specifies the treatment level for prediction.

`scores` calculates the score variables. Parameter-level scores are computed for the treatment mean and average treatment-effect equations. Equation-level scores are computed for the outcome equations.

The jth new variable will contain the scores for the jth parameter in the coefficient table if $j \leq t$, where t is the number of treatment levels. Otherwise, it will contain the scores for fitted equation $j-t$ following the first t parameters in the coefficient table.

Options for predict after stteffects wra

Main

`te`, the default, calculates the treatment effect for each noncontrol treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

`cmean` calculates the conditional mean for each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`censurv` calculates the survivor probability from the time-to-censoring model. (In other words, it calculates the probability that an outcome is not censored.) This option is allowed only if a censoring model is specified at estimation time. You need to specify only one new variable.

`xb` calculates the outcome model linear prediction at each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`lnshape` calculates the log of the conditional latent shape parameter for each treatment level or the treatment level specified in `tlevel()`. This option is valid when the outcome distribution Weibull, log normal, or gamma is used. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`clnshape` calculates the log of the conditional latent shape parameter of the censoring distribution. This option is valid when the censoring distribution Weibull, log normal, or gamma is used. You need to specify only one new variable.

`tlevel(`*treat_level*`)` specifies the treatment level for prediction.

`scores` calculates the score variables. Parameter-level scores are computed for the treatment mean and average treatment-effect equations. Equation-level scores are computed for the outcome and censoring equations.

The jth new variable will contain the scores for the jth parameter in the coefficient table if $j \leq t$, where t is the number of treatment levels. Otherwise, it will contain the scores for fitted equation $j-t$ following the first t parameters in the coefficient table.

Remarks and examples

Checking model specification is the most frequent reason for postestimation computation after stteffects. teoverlap provides a graphical method for checking the overlap assumption; see [CAUSAL] **teoverlap**. Summarizing the estimated probabilities provides another check. Recall that the reciprocals of these estimated probabilities are used as weights by some of the estimators. If the estimated probabilities are too small, the weights get too large and the estimators become unstable.

We estimate the average treatment effect of smoking on the time to a second heart attack by inverse-probability weighting; see example 1 of [CAUSAL] **stteffects ipw** for background.

```
. use https://www.stata-press.com/data/r19/sheart
(Time to second heart attack (fictional))
. stteffects ipw (smoke age exercise education) (age exercise diet education)
        Failure _d: fail
  Analysis time _t: atime
Iteration 0:  EE criterion = 2.042e-18
Iteration 1:  EE criterion = 1.890e-30
Survival treatment-effects estimation           Number of obs     =      2,000
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: logit
Censoring model: Weibull
```

_t	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
ATE						
smoke (Smoker vs Nonsmoker)	-2.22226	.6307573	-3.52	0.000	-3.458522	-.9859983
POmean						
smoke Nonsmoker	4.235569	.5210937	8.13	0.000	3.214244	5.256894

Below, we compute the estimated probabilities of being a Nonsmoker and store them in ps0. Likewise, the estimated probabilities of being a Smoker are stored in ps1.

```
. predict ps0 ps1, ps
```

The overlap condition requires that each of these probabilities be sufficiently greater than 0 and less than 1 for every individual; see *Assumptions and tradeoffs* under *Remarks and examples* in [CAUSAL] **stteffects intro**.

In practice, we know that weighting estimators perform poorly when the weights become too large. This approach requires that the probability of being a Nonsmoker not be too small among Nonsmokers and that the probability of being a Smoker not be too small among Smokers. Below, we summarize these probabilities.

```
. summarize ps0 if fail==1 & smoke==0
    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
         ps0 |        716    .6712529     .138754   .3872543   .9840293
. summarize ps1 if fail==1 & smoke==1
    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
         ps1 |        492    .4101277    .1101277   .0850604   .6125538
```

The minimum probability of being a `Nonsmoker` among `Nonsmokers` is 0.39. The minimum probability of being a `Smoker` among `Smokers` is 0.09. Neither minimum seems too small.

Estimating survival-time treatment effects also uses weights to adjust for censored outcomes; see [CAUSAL] **stteffects intro**. Thus we require that the probability of an uncensored failure also be sufficiently greater than 0. Below, we compute the estimated probabilities of failure and summarize them among those that fail.

```
. predict fprob2, censurv
. summarize fprob if fail==1
    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
      fprob2 |      1,208    .7246067    .2143543   .0364246   .9999086
```

The minimum probability of 0.04 does not appear too small.

❑ Technical note

The previous discussion builds on the intuition that the weights used in a weighting estimator should not be too large.

This technical note goes a little further by explicitly computing the weights and using them to replicate the inverse-probability-weighted point estimate for the `Nonsmoker` potential-outcome mean.

We now compute the weights using the predicted probabilities computed in the examples above and then use `mean` to compute the weighted average that estimates the potential-outcome mean for `Nonsmokers`.

```
. generate double ipw0 = 1/(ps0*fprob)
. mean _t [pw=ipw0] if smoke==0 & fail==1
Mean estimation                           Number of obs = 716
------------------------------------------------------------------------------
             |       Mean   Std. err.      [95% conf. interval]
-------------+----------------------------------------------------------------
          _t |   4.235569   .5820212       3.092894    5.378244
------------------------------------------------------------------------------
```

The weights account for data lost to the `Smoker` potential outcome or to censoring by increasing the importance of observations that were observed to be `Nonsmoker` failure times even though they were not likely to be observed.

The point estimate matches that reported by `stteffects ipw`; the standard errors differ because `mean` takes the estimated weights as given. See *Inverse-probability-weighted estimators* under *Methods and formulas* in [CAUSAL] **stteffects ipwra**.

❑

References

Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.

Bai, X., A. A. Tsiatis, and S. M. O'Brien. 2013. Doubly robust estimators of treatment-specific survival distributions in observational studies with stratified sampling. *Biometrics* 69: 830–839. https://doi.org/10.1111/biom.12076.

Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.

Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. https://doi.org/10.1016/j.jeconom.2009.09.023.

Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.

Guo, S., and M. W. Fraser. 2015. *Propensity Score Analysis: Statistical Methods and Applications*. 2nd ed. Thousand Oaks, CA: Sage.

Imbens, G. W. 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics* 86: 4–29. https://doi.org/10.1162/003465304323023651.

Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. https://doi.org/10.1257/jel.47.1.5.

Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55. https://doi.org/10.2307/2335942.

Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701. https://doi.org/10.1037/h0037350.

Tsiatis, A. A. 2006. *Semiparametric Theory and Missing Data*. New York: Springer.

Vittinghoff, E., D. V. Glidden, S. C. Shiboski, and C. E. McCulloch. 2012. *Regression Methods in Biostatistics: Linear, Logistic, Survival, and Repeated Measures Models*. 2nd ed. New York: Springer.

Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

[CAUSAL] **tebalance** — Check balance after teffects or stteffects estimation

[CAUSAL] **teoverlap** — Overlap plots

[CAUSAL] **stteffects ipw** — Survival-time inverse-probability weighting

[CAUSAL] **stteffects ipwra** — Survival-time inverse-probability-weighted regression adjustment

[CAUSAL] **stteffects ra** — Survival-time regression adjustment

[CAUSAL] **stteffects wra** — Survival-time weighted regression adjustment

[U] 20 Estimation and postestimation commands