cate postestimation — Postestimation tools for cate

Postestimation commands

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

Command	Description
`estat heterogeneity`	test for treatment-effects heterogeneity
`estat gatetest`	test for group treatment-effects heterogeneity
`estat ate`	compute the average treatment effect (ATE) for a subpopulation
`estat projection`	fit a linear projection of the individualized average treatment effect (IATE) estimates on variables
`estat series`	fit a nonparametric series regression of the IATE estimates on variables
`estat policyeval`	evaluate treatment-assignment policy
`estat tassigneval`	synonym of `estat policyeval`
`estat classification`	perform classification analysis of the data-driven groups
`categraph histogram`	histogram of the IATE predictions
`categraph gateplot`	plot of the group average treatment effect (GATE) or sorted GATE (GATES) estimates
`categraph iateplot`	plot of the IATE function estimates

The following postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
`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
`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`	predict the IATE function or its confidence intervals
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

Description for predict

`predict` predicts the observation-level IATE function, the standard errors, or the IATE confidence intervals.

Menu for predict

Statistics > Postestimation

Syntax for predict

Syntax for predicting the IATE or the standard errors

`predict` [*type*] *newvar* [*if*] [*in*] [`, iate stdp`]

Syntax for predicting the IATE confidence intervals

`predict` [*type*] $newvar_{ll}$ [*type*] $newvar_{ul}$ [*if*] [*in*], `ci` [`level(#)`]

$newvar_{ll}$ and $newvar_{ul}$ specify new variables for the lower and upper bounds of confidence intervals, respectively.

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

Options for predict

`iate`, the default, predicts the IATE function point estimates for each observation. The prediction can be from either a random forest or a parametric regression, which depends on the specification of `cmethod()` in `cate`. If the `cmethod(rforest)` option is specified in `cate`, the IATE prediction is computed using the generalized random forest. If the `cmethod(regress)` option is specified in `cate`, the IATE prediction is computed using a parametric linear regression. By default, a random forest prediction of the IATE function is computed.

`stdp` predicts the standard errors of the predictions of the IATE function. For the IATE predictions based on random forest, the standard errors are computed using a bootstrap of little bags, and for the IATE predictions based on linear regression, they are computed using the parametric delta method.

`ci` predicts the confidence intervals of the predictions of the IATE function.

`level(#)`, available only with `ci`, sets the confidence levels of the confidence intervals; the default is `level(95)`.

estat

Description for estat

`estat heterogeneity` tests the null hypothesis that the treatment effects are homogeneous.

`estat gatetest` tests the null hypothesis that the ATEs are the same among the specified subgroups. This command is allowed only when the `group()` option is specified in `cate`.

`estat ate` computes the ATE for a subpopulation defined by an `if` or `in` qualifier.

`estat projection` fits a linear projection of the estimated IATE function on specified variables.

`estat series` performs nonparametric series regression of the estimated IATE function on specified variables using a B-spline, piecewise polynomial spline, or polynomial basis.

`estat policyeval` evaluates the prespecified treatment-assignment rule. In particular, it computes the value of the policy and compares the difference of the two policies' values if two policies are specified.

`estat tassigneval` is a synonym of `estat policyeval`.

`estat classification` performs a classification analysis of the groups constructed based on the sorted IATE estimates. It performs a two-sample t test to compare the mean of a variable between the group with the largest treatment effects and the group with the smallest treatment effects. It is only allowed when the `group()` option is specified in `cate`.

Menu for estat

Statistics > Postestimation

Syntax for estat

Perform test of treatment-effects heterogeneity

`estat heterogeneity`

Perform test of group treatment-effects heterogeneity

`estat gatetest` [*grnumlist*] [, *gatetest_option*]

grnumlist is a numlist that specifies the group levels to be tested. If none are specified, all levels are used.

Compute the ATE for a subpopulation

`estat ate` [*if*] [*in*] [, *ate_options*]

Fit a linear projection of the IATE estimates on variables

`estat projection` [*varlist*] [*if*] [*in*] [, *projection_options*]

If *varlist* is not specified, then *catevarlist* specified in `cate` will be used.

Fit a nonparametric series regression of the IATE estimates on variables

`estat series` *indepvars*$_{\text{series}}$ [*if*] [*in*] [, *series_options*]

indepvars$_{\text{series}}$ is a list of independent variables for which a basis function will be formed.

Evaluate treatment-assignment policy

`estat policyeval` *policyvar*$_1$ [*policyvar*$_2$] [*if*] [*in*] [, *policy_options*]

policyvar$_1$ and *policyvar*$_2$ are variables specifying the probability of assigning each observation to treatment. If two *policyvar*s are specified, `estat policyeval` computes the values of each policy and their difference.

Perform classification analysis of the data-driven groups

`estat classification` *varname* [*if*] [*in*] [, *classification_options*]

gatetest_option	Description
`mtest`[`(`*mtest_options*`)`]	test each condition separately

mtest_options	Description
`noadjust`	no adjustment is to be made; the default
`bonferroni`	Bonferroni's method
`holm`	Holm's method
`sidak`	Šidák's method

Specifying `mtest` without an argument is equivalent to `mtest(noadjust)`.

ate_options	Description
`level(`#`)`	set confidence level; default is `level(95)`
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

projection_options	Description
`vce(`*vcetype*`)`	*vcetype* can be one of `ols` or `robust`
`level(`#`)`	set confidence level; default is `level(95)`
`noconstant`	suppress the constant term
`post`	post the results as the estimation results
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

series_options	Description
Model	
`bspline`	use a third-order B-spline basis; the default
`bspline(#)`	use a B-spline basis of order #
`spline`	use a third-order piecewise polynomial spline basis
`spline(#)`	use a piecewise polynomial spline basis of order #
`polynomial`	use a polynomial basis
`polynomial(#)`	use a polynomial basis of order #
`asis(`*varlist*`)`	include *varlist* in model as specified; do not use in basis
`nointeract(`*seriesvarlist*`)`	use *seriesvarlist* in basis without interactions
`criterion(`*crittype*`)`	criterion to use; *crittype* may be `cv`, `gcv`, `aic`, `bic`, or `mallows`
`knots(#)`	use a piecewise polynomial spline or B-spline basis function with # knots
`knotsmat(`*matname*`)`	use knots in matrix matname for piecewise polynomial spline or B-spline estimation
`distinct(#)`	minimum number of distinct values allowed in continuous covariates; default is `distinct(10)`
SE	
`vce(`*vcetype*`)`	*vcetype* can be one of `ols` or `robust`; default is `vce(robust)`
Graph	
* `graph`[`(`*seriesgraph_opts*`)`]	plot the prediction of conditional ATEs (CATEs)
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`aequations`	display auxiliary regression coefficients
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

* When `graph` or `graph()` is specified, only one variable is allowed in $indepvars_{series}$.

seriesgraph_opts	Description
`noci`	do not plot the confidence intervals
`cateopts(`*scatter_opts*`)`	affect rendition of the predicted CATE point estimates
`ciopts(`*area_opts*`)`	affect rendition of the confidence interval
twoway_options	any options other than `by()` documented in [G-3] ***twoway_options***

scatter_opts	Description
connect_options	change the look of lines or connecting method
marker_options	change the look of markers (color, size, etc.)

policy_options	Description
`level(#)`	set confidence level; default is `level(95)`
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

classification_options	Description
unequal	data have unequal variances
welch	use Welch's approximation
level(#)	set confidence level; default is level(95)

Options for estat

Options for estat are presented under the following headings:

Options for estat gatetest

mtest[(*mtest_options*)] specifies that tests be performed for each condition separately. *mtest_options* specifies the method for adjusting *p*-values for multiple tests and can be the following:

noadjust specifies that no adjustment is to be made.

bonferroni specifies that Bonferroni's method be used.

holm specifies that Holm's method be used.

sidak specifies that Šidák's method be used.

Specifying mtest without an argument is equivalent to specifying mtest(noadjust).

Options for estat ate

level(#); see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Options for estat projection

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (ols) or that are robust to some kinds of misspecification (robust); see [R] ***vce_option***. The default is vce(robust).

vce(ols) uses the standard variance estimator for ordinary least-squares regression.

level(#); see [R] **Estimation options**.

noconstant suppresses the constant term.

post posts the results as the estimation results, so all postestimation commands after regress will be available; see [R] **regress postestimation**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Options for estat series

Model

bspline specifies that a third-order B-spline be selected as the basis. It is the default basis.

bspline(#) specifies that a B-spline of order # be used as the basis. The order may be 1, 2, or 3.

spline specifies that a third-order piecewise polynomial spline be selected as the basis.

spline(#) specifies that a piecewise polynomial spline of order # be used as the basis. The order may be 1, 2, or 3.

polynomial specifies that a polynomial be selected as the basis.

polynomial(#) specifies that a polynomial of order # be used as the basis. The order may be an integer between 1 and 16.

asis(*varlist*) specifies that variables in *varlist* be included as independent variables in the model without any transformation. No B-spline, piecewise polynomial spline, or polynomial basis function will be formed from these variables. Variables in *varlist* may not be specified in $indepvars_{series}$.

nointeract(*seriesvarlist*) specifies that the terms in the basis function formed from variables in *seriesvarlist* not be interacted with the terms of the basis function formed from other variables in $indepvars_{series}$. Covariates specified in *seriesvarlist* must be in $indepvars_{series}$.

criterion(*crittype*) specifies that *crittype* be used to select the optimal number of terms in the basis function. *crittype* may be one of the following: cv (cross-validation), gcv (generalized cross-validation), aic (Akaike's information criterion), bic (Schwarz's Bayesian information criterion), or mallows (Mallows's C_p). The default is criterion(cv).

knots(#) specifies that a piecewise polynomial spline or B-spline basis function with # knots be used. The minimum number of knots must be an integer greater than or equal to 1. The maximum number of knots is either 4,096 or two-thirds of the sample size, whichever is smaller.

knotsmat(*matname*) specifies that the knots for each continuous covariate be the values in each row of *matname*. The number of knots should be the same for each covariate, and there must be as many rows as there are continuous covariates. If rows of *matname* are not labeled with varnames, then rows are assumed to be in the order of $indepvars_{series}$.

distinct(#) specifies the minimum number of distinct values allowed in continuous variables. By default, continuous variables that enter the basis through either $indepvars_{series}$ or *seriesvarlist* are required to have at least 10 distinct values. Continuous variables with few distinct values provide little information for determining an appropriate basis function and may produce unreliable estimates.

SE

vce(*vcetype*) specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (robust) and that assume homoskedasticity (ols); see [R] ***vce_option***. The default is vce(robust).

Graph

graph[(*seriesgraph_opts*)] plots the prediction of CATEs. *seriesgraph_opts* may be the following:

noci specifies not to plot the confidence intervals.

cateopts(*scatter_opts*) affects the rendition of the predicted CATE point estimates. *scatter_opts* may be the following:

connect_options specify how points on a graph are to be connected; see [G-3] ***connect_options***.

marker_options affect the rendition of markers drawn at the plotted points, including their shape, size, color, and outline; see [G-3] ***marker_options***.

ciopts(*area_options*) affects the rendition of the confidence intervals; see [G-3] ***area_options***.

twoway_options are any of the options documented in [G-3] ***twoway_options***, excluding by(). These include options for titling the graph (see [G-3] ***title_options***) and for saving the graph to disk (see [G-3] ***saving_option***).

When graph or graph() is specified, only one variable is allowed in $indepvars_{series}$.

Reporting

level(#); see [R] **Estimation options**.

aequations specifies that the auxiliary regression coefficients be reported. By default, only the average marginal effects of the covariates on the treatment effects are reported.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Options for estat policyeval

level(#); see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

Options for estat classification

unequal specifies that the unpaired data not be assumed to have equal variances.

welch specifies that the approximate degrees of freedom for the test be obtained from Welch's (1947) formula rather than from Satterthwaite's (1946) approximation formula, which is the default when unequal is specified. Specifying welch implies unequal.

level(#); see [R] **Estimation options**.

categraph

Description for categraph

`categraph histogram` plots the histogram of the IATE predictions.

`categraph gateplot` plots the GATE or GATES estimates and their confidence intervals.

`categraph iateplot` plots the IATE function and the pointwise confidence intervals when one variable is varying and the other variables are fixed at specific values.

Menu for categraph

Statistics > Postestimation

Syntax for categraph

Histogram of the IATE predictions

`categraph histogram` [*if*] [*in*] [, *histogram_options*]

Plot of the GATE or GATES estimates

`categraph gateplot` [, *gateplot_options*]

Plot of the IATE function estimates

`categraph iateplot` *xvar* [*if*] [*in*] [, *iateplot_options*]

xvar is a variable name specified in *catevarlist* of `cate`. It can be a regular variable or a factor variable, but interaction and product notations are not allowed.

gateplot_options	Description
Main	
`level(#)`	set confidence level; default is `level(95)`
`noci`	do not plot the confidence intervals
Scatter options	
`gateopts(`*scatter_opts*`)`	affect rendition of the predicted GATE or GATES point estimates
CI options	
`ciopts(`*area_opts*`)`	affect rendition of the confidence interval
Y axis, X axis, Titles, Legend, Overall	
twoway_options	any options other than `by()` documented in [G-3] ***twoway_options***

scatter_opts	Description
connect_options	change the look of lines or connecting method
marker_options	change the look of markers (color, size, etc.)

iateplot_options	Description
Model	
* `range(`*#_min #_max*`)`	plot IATE function over *xvar* equal to *#_min* to *#_max*; the default sets minimum and maximum of *xvar* in the current dataset
* `range(`*varname*`)`	plot IATE function over *xvar* equal to minimum and maximum of *varname*
* `n(`#`)`	evaluate at # points; default is 300 points
`level(`#`)`	set confidence level; default is `level(95)`
`at(`*atspec*`)`	set values for all *catevarlist* except *xvar*
IATE plot	
`iateopts(`*scatter_opts*`)`	affect rendition of the predicted IATE point estimates
CI	
`noci`	do not plot the confidence intervals
`ciopts(`*area_opts*`)`	affect rendition of the confidence interval
Y axis, X axis, Titles, Legend, Overall	
twoway_options	any options other than `by()` documented in [G-3] ***twoway_options***

*`range()` and `n()` are not allowed if *xvar* is a factor variable in *catevarlist* of `cate`.

Options for categraph

Options for `categraph` are presented under the following headings:

Options for categraph histogram
Options for categraph gateplot
Options for categraph iateplot

Options for categraph histogram

histogram_options are options in [R] **histogram**.

Options for categraph gateplot

Main

`level(`#`)`; see [R] **Estimation options**.

`noci` specifies not to plot the confidence intervals.

Scatter options

`gateopts(`*scatter_opts*`)` affects the rendition of the predicted GATE or GATES point estimates. *scatter_opts* may be the following:

connect_options specify how points on a graph are to be connected; see [G-3] ***connect_options***.

marker_options affect the rendition of markers drawn at the plotted points, including their shape, size, color, and outline; see [G-3] ***marker_options***.

CI options

`ciopts(`*area_options*`)` affects the rendition of the confidence intervals; see [G-3] ***area_options***.

Y axis, X axis, Titles, Legend, Overall

twoway_options are any of the options documented in [G-3] ***twoway_options***, excluding `by()`. These include options for titling the graph (see [G-3] ***title_options***) and for saving the graph to disk (see [G-3] ***saving_option***).

Options for categraph iateplot

Model

`range(`#*_min* #*_max*`)` or `range(`*varname*`)` plots the IATE function over *xvar* between #*_min* and #*_max* or between the minimum or maximum of *varname*, respectively, while holding other variables in *catevarlist* of `cate` fixed at some values. The default sets #*_min* and #*_max* to the minimum and maximum of *xvar* in the current dataset. See `at()` below for details on fixing values for the variables other than *xvar*.

`n(`#`)` evaluates the IATE function at # points. The points on *xvar* are evenly spaced between the minimum and the maximum specified in `range()`. The other variables in *catevarlist* are fixed at some values specified in `at()`. The default is `n(300)`.

`level(`#`)`; see [R] **Estimation options**.

`at(`*atspec*`)` specifies values for all the covariates (except *xvar*) in *catevarlist* of `cate` to be treated as fixed.

atspec may contain one or more of the following specifications:

(*stat*) *varlist*

varname = #

where

1. Variable names (whether in *varlist* or *varname*) must be the covariates in *catevarlist* other than *xvar* in the `cate` estimation.
2. Variable names may be continuous variables or factor variables.
3. *varlist* may also be one of three standard lists:
   (a) `_all` (all covariates),
   (b) `_factor` (all factor-variable covariates), or
   (c) `_continuous` (all continuous covariates).

4. *stat* may be any of the following:

stat	Description	Variables allowed
`mean`	means (default for continuous variables)	continuous
`base`	base level (default for factor variables)	factors
`median`	medians	continuous
`p1`	1st percentile	continuous
`p2`	2nd percentile	continuous
...	3rd–49th percentiles	continuous
`p50`	50th percentile (same as `median`)	continuous
...	51st–97th percentiles	continuous
`p98`	98th percentile	continuous
`p99`	99th percentile	continuous
`min`	minimums	continuous
`max`	maximums	continuous
`zero`	fixed at zero	continuous

IATE plot

`iateopts(`*scatter_opts*`)` affects the rendition of the predicted IATE point estimates. *scatter_opts* may be the following:

connect_options specify how points on a graph are to be connected; [G-3] ***connect_options***.

marker_options affect the rendition of markers drawn at the plotted points, including their shape, size, color, and outline; see [G-3] ***marker_options***.

CI

`noci` specifies not to plot the confidence intervals.

`ciopts(`*area_options*`)` affects the rendition of the confidence intervals; see [G-3] ***area_options***.

Y axis, X axis, Titles, Legend, Overall

twoway_options are any of the options documented in [G-3] ***twoway_options***, excluding `by()`. These include options for titling the graph (see [G-3] ***title_options***) and for saving the graph to disk (see [G-3] ***saving_option***).

Remarks and examples

For an overview of `cate` postestimation tools and the examples that demonstrate how to use the `cate` command and its postestimation tools, see details in *Remarks and examples* in [CAUSAL] **cate**.

The `estimates` commands after the `cate` command work the same as they do after other estimation commands with only one difference: `estimates save` *filename* saves three files, not just one. *filename*`.ster`, *filename*`.stgrf`, and *filename*`.stxer` are saved. See [R] **estimates** for details.

Stored results

`estat heterogeneity` stores the following in `r()`:

Scalars

`r(p)`	two-sided p-value
`r(df)`	test constraints degrees of freedom
`r(chi2)`	χ^2

Matrices

`r(b)`	coefficient vector in the best linear prediction of IATE
`r(V)`	variance–covariance matrix of the estimators

`estat gatetest` stores the following in `r()`:

Scalars

`r(p)`	two-sided p-value
`r(df)`	test constraints degrees of freedom
`r(chi2)`	χ^2
`r(drop)`	`1` if constraints were dropped, `0` otherwise

`estat ate` stores the following in `r()`:

Scalars

`r(N)`	number of observations

Matrices

`r(b)`	coefficient vector
`r(V)`	variance–covariance matrix of the estimators
`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

`estat projection` stores the following in `r()`:

Scalars

`r(N)`	number of observations
`r(mss)`	model sum of squares
`r(df_m)`	model degrees of freedom
`r(rss)`	residual sum of squares
`r(df_r)`	residual degrees of freedom
`r(r2)`	R^2
`r(r2_a)`	adjusted R^2
`r(F)`	F statistic
`r(rmse)`	root mean squared error
`r(ll)`	log likelihood under additional assumption of independent and identically distributed normal errors
`r(ll_0)`	log likelihood, constant-only model
`r(rank)`	rank of `r(V)`

Matrices

`r(b)`	coefficient vector
`r(V)`	variance–covariance matrix of the estimators
`r(beta)`	standardized coefficients
`r(V_modelbased)`	model-based variance
`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

estat projection with the post option stores the following in e():

Scalars

e(N)	number of observations
e(mss)	model sum of squares
e(df_m)	model degrees of freedom
e(rss)	residual sum of squares
e(df_r)	residual degrees of freedom
e(r2)	R^2
e(r2_a)	adjusted R^2
e(F)	F statistic
e(rmse)	root mean squared error
e(ll)	log likelihood under additional assumption of i.i.d. normal errors
e(ll_0)	log likelihood, constant-only model
e(rank)	rank of e(V)

Macros

e(cmd)	regress
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(model)	ols
e(title)	title in estimation output when vce() is not ols
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict
e(marginsok)	predictions allowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators
e(beta)	standardized coefficients
e(V_modelbased)	model-based variance

Functions

e(sample)	marks estimation sample

estat series stores the following in r():

Scalars

r(N)	number of observations
r(converged)	1 if converged, 0 otherwise
r(order)	order of basis function
r(rank)	rank of r(V)

Matrices

r(b)	coefficient vector
r(V)	variance–covariance matrix of the estimators
r(V_modelbased)	model-based variance
r(ilog)	iteration log (up to 20 iterations)
r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

`estat policyeval` stores the following in `r()`:

Scalars
- `r(N)` number of observations

Macros
- `r(policy_var1)` first policy variable name
- `r(policy_var2)` second policy variable name

Matrices
- `r(b)` coefficient vector
- `r(V)` variance–covariance matrix of the estimators
- `r(table)` matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

`estat classification` stores the following in `r()`:

Scalars
- `r(N_1)` sample size n_1
- `r(N_2)` sample size n_2
- `r(p_l)` lower one-sided p-value
- `r(p_u)` upper one-sided p-value
- `r(p)` two-sided p-value
- `r(se)` estimate of standard error
- `r(t)` t statistic
- `r(sd_1)` standard deviation for population 1
- `r(sd_2)` standard deviation for population 2
- `r(sd)` combined standard deviation
- `r(mu_1)` $\overline{x}_1$ mean for population 1
- `r(mu_2)` $\overline{x}_2$ mean for population 2
- `r(df_t)` degrees of freedom
- `r(level)` confidence level

`categraph iateplot` stores the following in `r()`:

Macros
- `r(xvar)` variable allowed to vary
- `r(vtype_list)` types of variables other than `r(xvar)`
- `r(vname_list)` names of variables other than `r(xvar)`
- `r(stat_list)` statistics of variables other than `r(xvar)`

Matrices
- `r(at)` matrix of values from the `at()` options

Methods and formulas

Methods and formulas are presented under the following headings:

For notational simplicity, we drop the subscript i indicating the ith observation to refer to a random variable.

IATE predictions

`predict` predicts the IATEs, their standard errors, or the IATE confidence intervals. The IATEs can be estimated by either a generalized random forest or a parametric linear regression, which is specified in the `cmethod()` option of `cate`. For details of the random-forest-based IATE predictions, their standard errors, and the confidence intervals, see *Generalized random forest* in [CAUSAL] **cate**. For details on linear-regression-based predictions and their standard errors, see the discussions in *Methods and formulas* of [R] **predict**; the confidence intervals are obtained via the delta method.

Test of treatment-effects heterogeneity

`estat heterogeneity` tests the null hypothesis that the treatment effects are homogeneous. In particular, it implements the test proposed in Chernozhukov et al. (2006). Let $\tau_0(\mathbf{x})$ be the true IATE function, $\hat{\tau}(\mathbf{x})$ be an estimate of the IATE function, $\bar{\tau}$ be $\mathbf{E}\{\hat{\tau}(\mathbf{x})\}$, and $\check{\tau}$ be the sample average of $\hat{\tau}(\mathbf{x})$. Then the best linear prediction of $\tau_0(\mathbf{x})$ conditional on $\hat{\tau}(\mathbf{x})$ is given by

$$\tau_0(\mathbf{x}) = \gamma_1\bar{\tau} + \gamma_2\{\hat{\tau}(\mathbf{x}) - \bar{\tau}\} + \epsilon$$

where ϵ is the error term.

If $\gamma_2 = 0$, it implies that the $\hat{\tau}(\mathbf{x})$ predictions are pure noise, and it also means that $\tau_0(\mathbf{x})$ is constant or homogeneous. Thus, to test the null hypothesis that the treatment effects are homogeneous, we perform a Wald test of $\gamma_2 = 0$.

In the partialing-out estimator, the coefficients of γ_1 and γ_2 can be identified by fitting the following regression:

$$y - \hat{h}(\mathbf{x},\mathbf{w}) = \gamma_1\check{\tau}\{d - \hat{f}(\mathbf{x},\mathbf{w})\} + \gamma_2\{\hat{\tau}(\mathbf{x}) - \check{\tau}\}\{d - \hat{f}(\mathbf{x},\mathbf{w})\} + \epsilon$$

where y is the outcome variable, $\hat{h}(\mathbf{x},\mathbf{w})$ estimates $\mathbf{E}(y|\mathbf{x},\mathbf{w})$, d is the treatment variable, and $\hat{f}(\mathbf{x},\mathbf{w})$ estimates $\mathbf{E}(d|\mathbf{x},\mathbf{w}) \equiv \mathbf{P}(d = 1|\mathbf{x},\mathbf{w})$.

In the augmented inverse-probability weighting (AIPW) estimator, the best linear prediction of $\tau_0(\mathbf{x})$ conditional on $\hat{\tau}(\mathbf{x})$ can be obtained by regressing the AIPW scores implied by the full interactive model on $\check{\tau}$ and $\hat{\tau}(\mathbf{x}) - \check{\tau}$.

Test of group-level treatment-effects heterogeneity

`estat gatetest` tests the null hypothesis that the ATEs are the same among the specified subgroup levels. It performs Wald tests on the GATE estimates' coefficients. For details of Wald tests, see *Methods and formulas* in [R] **test**.

ATE for a subsample

`estat ate` computes the ATE for a subsample by taking the average of the AIPW scores implied by the model over the subsample, which is proposed in Chernozhukov et al. (2018) and Knaus (2022). For details of the AIPW scores in the partial linear and the fully interactive models, see *Methods and formulas* in [CAUSAL] **cate**.

Linear or nonparametric series projection of the IATE on variables

`estat projection` computes the linear projection of the IATE function on the specified variables. Similarly, `estat series` computes the nonparametric series projection of the IATE function on the basis functions formed by the specified variables. The linear projection is a special case of the series projection that uses the basis functions as the variables themselves. Thus, we only need to discuss the methods for `estat series`, because `estat projection` is just a special case.

`estat series` implements the methods proposed by Semenova and Chernozhukov (2021) by running a series regression of the AIPW scores implied by the model on the basis functions formed by the specified variables. For details of the AIPW scores in the partial linear and the fully interactive models, see *Methods and formulas* in [CAUSAL] **cate**. For a discussion of nonparametric series regression, see *Methods and formulas* in [R] **npregress series**.

Treatment-assignment policy evaluation

`estat policyeval` or `estat tassigneval` evaluates treatment-assignment policies. Suppose a treatment-assignment rule assigns individuals to be treated or not treated. We want to evaluate this treatment-assignment rule by answering questions such as the following:

1. If we implement such a rule, what is the average outcome of the population?

2. Furthermore, if we have two different rules, which is better?

For the first question, we compute the average of the outcome if the treatment is assigned according to a rule. We estimate

$$\Pi(\pi) = \mathbf{E}[\pi(\mathbf{x})y(1) + \{1 - \pi(\mathbf{x})\}\, y(0)]$$

where $y(1)$ is the potential outcome when it is treated, $y(0)$ is the potential outcome when it is not treated, and $\pi(\mathbf{x}) \in [0, 1]$ is a prespecified treatment-assignment probability, which is also known as a policy. $\Pi(\pi)$ is also called the value of the policy π.

For the second question, we compute the difference of the values between two policies, π_1 and π_2. In particular, we compute the contrast of the values between the two treatment-assignment policies.

$$\Pi(\pi_1) - \Pi(\pi_2)$$

For details of the potential outcomes in the partial linear and the fully interactive models, see *Methods and formulas* in [CAUSAL] **cate**.

Classification analysis

`estat classification` performs a classification analysis of the groups constructed based on the sorted IATE estimates. It performs a two-sample t test to compare the mean of a variable between the least and the most affected groups. For details of t tests on the equality of means, see *Methods and formulas* in [R] **ttest**.

References

Chernozhukov, V., D. Chetverikov, M. Demirer, E. Duflo, C. B. Hansen, W. K. Newey, and J. M. Robins. 2018. Double/debiased machine learning for treatment and structural parameters. *Econometrics Journal* 21: C1–C68. https://doi.org/10.1111/ectj.12097.

Chernozhukov, V., M. Demirer, E. Duflo, and I. Fernández-Val. 2006. Generic machine learning inference on heterogeneous treatment effects in randomized experiments, with an application to immunization in India. NBER Working Paper 24678, National Bureau of Economic Research. https://doi.org/10.3386/w24678.

Knaus, M. C. 2022. Double machine learning–based programme evaluation under unconfoundedness. *Econometrics Journal* 25: 602–627. https://doi.org/10.1093/ectj/utac015.

Satterthwaite, F. E. 1946. An approximate distribution of estimates of variance components. *Biometrics Bulletin* 2: 110–114. https://doi.org/10.2307/3002019.

Semenova, V., and V. Chernozhukov. 2021. Debiased machine learning of conditional average treatment effects and other causal functions. *Econometrics Journal* 24: 264–289. https://doi.org/10.1093/ectj/utaa027.

Welch, B. L. 1947. The generalization of 'student's' problem when several different population variances are involved. *Biometrika* 34: 28–35. https://doi.org/10.2307/2332510.

Also see

[CAUSAL] **cate** — Conditional average treatment-effects estimation

[U] 20 Estimation and postestimation commands