xtswitchdid postestimation — Postestimation tools for xtswitchdid[+]

[+]Postestimation features after `xtswitchdid` are part of StataNow.

Postestimation commands

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

Command	Description
`estat ptrends`	parallel-trends and no-anticipation test using placebos
`estat eventplot`	plot the event-study effects at each exposure
`estat total`	estimate average total effect
`estat paths`	tabulate treatment paths observed in the estimation sample

The following 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
`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
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

estat

Description for estat

`estat ptrends` tests the parallel-trends and no-anticipation assumptions by testing whether all available placebos are equal to zero and optionally displays the placebo estimates in a graph.

`estat eventplot` plots the event-study effects for each exposure.

`estat total` aggregates the event-study effects across exposures to give an average total effect.

`estat paths` tabulates the treatment paths observed in the estimation sample and generates a variable encoding the relevant treatment path for each group. It optionally produces indicator variables for each path, which can be used by the main command to produce path-specific estimates.

Menu for estat

Statistics > Postestimation

Syntax for estat

Test parallel-trends and no-anticipation assumptions using placebos

```
estat ptrends [ , ptrends_options ]
```

Plot event-study effects

```
estat eventplot [ , eventplot_options ]
```

Compute average total effect

```
estat total [ , level(#) ]
```

Tabulate treatment paths

```
estat paths [ , paths_options ]
```

ptrends_options	Description
`level(#)`	set confidence level; default is `level(95)`
`graph`	display a plot of placebo estimates
`graph(`*graph_opts*`)`	affect rendition of the aggregation plot
`alleffects`	plot main effects with placebo estimates

eventplot_options	Description
`level(#)`	set confidence level
graph_opts	affect rendition of the event-study effects plot

paths_options	Description
`pathlen(#)`	set path length
`freqsort`	request paths be listed in descending frequency order
`generate(`*pathstub*`)`	generate path indicator variables with names *pathstub*#
`pathvar(`*pvar*`[ , replace ])`	specify the name for the generated path variable

graph_opts	Description
Main	
`noci`	do not plot the confidence intervals
Marker options	
marker_options	change the look of markers (color, size, etc.)
Line options	
connect_options	change the look of lines or connecting method
CI plot	
`ciopts(`*area_options*`)`	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***

Options for estat

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

Options for estat ptrends

`level(#)` specifies the confidence level, as a percentage, for CIs. The default is `level(95)` or as set by `set level`; see **[U] 20.8 Specifying the width of confidence intervals**.

`graph` requests that a graph of the placebo estimates be displayed.

`graph(`*graph_opts*`)` affects the rendition of the placebo estimates plot. *graph_opts* may be the following:

`noci` removes plots of the CIs. The default is to plot the CIs.

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

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

`ciopts(`*area_options*`)` affects the rendition of the CIs; 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***).

`alleffects` specifies that the graph, if requested, plot the main event-study effects along with the placebo estimates.

Options for estat eventplot

`level(#)` specifies the confidence level, as a percentage, for CIs. The default is `level(95)` or as set by `set level`; see **[U] 20.8 Specifying the width of confidence intervals**.

Main

`noci` removes plots of the CIs. The default is to plot the CIs.

Marker 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***.

Line options

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

CI plot

`ciopts(`*area_options*`)` affects the rendition of the CIs; 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***).

Option for estat total

`level(#)` specifies the confidence level, as a percentage, for CIs. The default is `level(95)` or as set by `set level`; see **[U] 20.8 Specifying the width of confidence intervals**.

Options for estat paths

`pathlen(#)` specifies the length of the treatment paths to tabulate. By default, `estat paths` uses the number of event-study effects computed in the main command.

`freqsort` requests that the treatment paths be listed in descending frequency order. By default, they are listed in lexicographical order.

`generate(`*pathstub*`)` requests that indicator variables be generated for each treatment path. The new variables will be named *pathstub#*.

`pathvar(`*pvar*`[ , replace ])` specifies *pvar* as the name for the generated treatment path variable, which encodes each treatment path as a different number and assigns value labels for each path. By default, `_did_paths` is used as the name of this path variable. If `_did_paths` already exists in the dataset, it is replaced if `pathvar()` is not specified.

`replace` specifies that *pvar* be overwritten if it exists.

Remarks and examples

For examples of the `estat` commands above, see [CAUSAL] **xtswitchdid**.

Stored results

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

Scalars

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

Matrices

`r(b)`	coefficient vector
`r(V)`	variance–covariance matrix of the estimators
`r(table)`	matrix containing test statistics and critical values

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

Matrices

`r(table)`	matrix containing test statistics and critical values

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

Scalars

`r(b)`	coefficient
`r(V)`	variance of the estimator

Matrices

`r(table)`	matrix containing test statistics and critical values

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

Scalars

`r(n_paths)`	number of treatment paths
`r(pathlen)`	path length

Macros

`r(pathvar)`	name of path variable
`r(pathvar)`	stub name for path indicator variables

Methods and formulas

Methods and formulas are presented under the following headings:

Test for all placebos being zero

`estat ptrends` tests implications of the parallel-trends and no-anticipation assumptions by testing whether all observed placebos are zero, as described in de Chaisemartin, D'Haultfœuille, and Vazquez-Bare (2024).

Placebo estimates for the ℓth exposure are computed in the same way that event-study effect estimates are computed, but the observed outcome $y_{g,t_g+\ell}$ is replaced by the observed outcome $y_{g,t_g-\ell}$. In other words, we still take the last unswitched period as a baseline, but the difference is taken using the outcome ℓ periods before, rather than ℓ periods after, provided that outcome is available in the data. If the parallel-trends and no-anticipation assumptions hold, we expect these differences to be zero on average.

The normalized placebo estimator without covariates for the ℓth exposure is written, using notation similar to that in [CAUSAL] **xtswitchdid**, as

$$\hat{\delta}_\ell^{n,p} = \frac{1}{\delta_\ell^{d,p}} \sum_{g:t_g-\ell\geq 1, t_g+\ell\leq T_g} s_g \left\{ n_{g,t_g+\ell}(y_{g,t_g-\ell} - y_{g,t_g}) - \frac{1}{n^g_{t_g+\ell}} \sum_{\substack{g': d_{g'1}=d_{g1} \\ t_{g'}>t_g+\ell}} n_{g',t_g+\ell}(y_{g',t_g-\ell} - y_{g',t_g}) \right\}$$

where $\delta_{g,\ell}^{d,p} = \sum_{k=0}^{\ell-1}(d_{g,t_g-k} - d_{g,1})$ is the total "dose" of treatment received by a group g from t_g "until" $t_g - \ell$ and $\delta_\ell^{d,p}$ is the total of the absolute values of such doses over each group for which the ℓth placebo estimate can be computed.

Note that, as written, the ℓth placebo estimate can be computed only if the ℓth event-study effect estimate is computed. Only groups for which the ℓth event-study effect is available are included in the ℓth placebo estimator. This means that if L event-study effects have been computed, there can be at most L placebo estimates computed.

The raw placebo estimator $\hat{\delta}_\ell^p$ is identical to its normalized equivalents with δ_ℓ^d replaced by n_ℓ. Estimators with covariates are computed identically as for the main effects, with the same modifications as above.

The covariance estimator for the placebos also corresponds to the covariance estimator for the main effects, with future outcomes at a time period $t_g + \ell$ replaced by past outcomes at time period $t_g - \ell$.

The placebos are tested against the null of all estimates being zero using `test`. When the variance–covariance matrix is rank deficient, such as when there are more placebo estimates than clusters, constraints are dropped and a note is printed.

Measure of average total effect

For the sake of brevity, we restrict attention to when there are no switchers-out. See de Chaisemartin et al. (2025) for the computation of $\hat{\delta}$ in the general case, which involves constructing separate quantities for switchers-out and switchers-in and combining use of appropriate weights. `estat total` aggregates over exposures to compute the following effect,

$$\hat{\delta} = \frac{\sum_{g:t_g<T_g} \sum_{\ell=1}^{T_g-t_g} \hat{\delta}_{g,\ell}}{\sum_{g:t_g<T_g} \sum_{\ell=1}^{T_g-t_g} (d_{g,t_g+\ell} - d_{g,1})}$$

where

$$\hat{\delta}_{g,l} = n_{g,t_g+\ell}(y_{g,t_g+\ell} - y_{g,t_g}) - \frac{1}{n^g_{t_g+\ell}} \sum_{\substack{g': d_{g'1}=d_{g1} \\ t_{g'}>t_g+\ell}} n_{g',t_g+\ell}(y_{g',t_g+\ell} - y_{g',t_g})$$

When the treatment is absorbing, this effect can be interpreted as an average effect of the total treatment dose over each period.

In the case with no switchers-out we consider here, $\hat{\delta}$ can be written as a weighted average of the raw estimators $\hat{\delta}_\ell$ over exposures. The variance estimator for $\hat{\delta}$ can be computed accordingly. See de Chaisemartin et al. (2025) for the general case.

References

de Chaisemartin, C., D. Ciccia, F. Knau, M. Malézieux, D. Sow, D. Arboleda, R. Angotti, X. D'Haultfœuille, B. Li, H. Fabre, and A. Quispe. 2025. Using did_multiplegt_dyn to estimate event-study effects in complex designs: Overview, and four examples based on real datasets. Unpublished manuscript. https://doi.org/10.2139/ssrn.5337463.

de Chaisemartin, C., X. D'Haultfœuille, and G. Vazquez-Bare. 2024. Difference-in-difference estimators with continuous treatments and no stayers. *American Economic Association Papers and Proceedings* 114: 610–613. https://doi.org/10.1257/pandp.20241049.

Also see

[CAUSAL] **xtswitchdid** — Difference in differences with switching treatments for panel data+

[CAUSAL] **DID intro** — Introduction to difference-in-differences estimation

[U] 20 Estimation and postestimation commands