

[Description](#)[Options](#)[Also see](#)[Quick start](#)[Remarks and examples](#)[Menu](#)[Stored results](#)[Syntax](#)[References](#)

Description

`tebalance overrides` performs a test for covariate balance after estimation by a `teffects` inverse-probability-weighted (IPW) estimator or an `stteffects` IPW estimator.

Quick start

Test for covariate balance after a `teffects` or an `stteffects` IPW estimator

```
tebalance overrides
```

Same as above, but test for balance only in base covariates and exclude interaction terms

```
tebalance overrides, bonly
```

Menu

Statistics > Causal inference/treatment effects > Balance > Overidentification test

Syntax

```
tebalance overid [ , bconly nolog iterate(#) ]
```

collect is allowed; see [U] [11.1.10 Prefix commands](#).

Options

Main

`bconly` specifies that only the base covariates be included in the test for balance. By default, the powers and interactions specified by factor-variable notation in the `teffects` or `stteffects` model are also included in the test for balance.

`nolog` suppresses the display of the optimization search log.

`iterate(#)` sets the maximum number of iterations to `#` in the generalized method of moments estimator used to compute the test statistic.

Remarks and examples

When the distribution of a covariate is the same for all treatment levels, the covariate is said to be balanced. `tebalance overid` performs a test to see whether the covariates are balanced after `teffects` or `stteffects`. `tebalance overid` can be executed after `teffects ipw`, `teffects aipw`, `teffects ipwra`, `stteffects ipw`, or `stteffects ipwra`, which use the inverse-probability weights predicted by a treatment model to account for how treatment assignment depends on observed covariates. If the treatment model is well specified, IPW functions of the covariates from the model are balanced.

We recommend that you read [\[CAUSAL\] tebalance](#) before proceeding; it provides an introduction to covariate balance and an overview of the implemented methods. See [\[CAUSAL\] stteffects intro](#) for survival-time discussion and examples.

► Example 1: Base covariates and interactions

This example illustrates the interpretation of the `bconly` option, which excludes powers and interactions when factor variables are included in the propensity-score model.

We frequently use factor variables to include powers of, and interactions between, base covariates in our specification of the propensity-score model. In [example 1](#) under *Remarks and examples* in [\[CAUSAL\] tebalance](#), we rejected the null hypothesis of balance in a model using only base covariates but not in the richer model that included power and interaction terms. By default, `tebalance overid` tests whether the model balances the base covariates and the power-and-interaction covariates. When option `bconly` is specified, `tebalance overid` tests whether the model balances the base covariates only.

Using an extract from the data used by [Cattaneo \(2010\)](#), we use `teffects ipw` to estimate the effect of a mother's smoking behavior (`mbsmoke`) on the birthweight of her child (`bweight`), controlling for marital status (`mmarried`), the mother's age (`mage`), whether the mother had a prenatal doctor's visit in the baby's first trimester (`prenatal1`), and whether this baby is the mother's first child (`fbaby`). In addition to the base covariates, we include the square of `mage`, an interaction between `mage` and `mmarried`, and an interaction between `mage` and `prenatal1` in the model for the propensity score.

```

. use https://www.stata-press.com/data/r19/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ipw (bweight) (mbsmoke mmarried mage prenatal1 fbaby
> c.mage#(c.mage i.mmarried prenatal1)), aequations
Iteration 0: EE criterion = 9.365e-20
Iteration 1: EE criterion = 2.612e-26

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: logit

```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
mbsmoke (Smoker vs Nonsmoker)	-239.6875	26.43427	-9.07	0.000	-291.4977	-187.8773
POmean						
mbsmoke Nonsmoker	3403.638	9.56792	355.73	0.000	3384.885	3422.39
TME1						
mmarried	.8522468	.462536	1.84	0.065	-.0543072	1.758801
mage	.1742823	.0651039	2.68	0.007	.0466811	.3018836
prenatal1	.4018114	.4341762	0.93	0.355	-.4491584	1.252781
fbaby	-.4824413	.0868982	-5.55	0.000	-.6527587	-.3121239
c.mage# c.mage	-.002515	.0012585	-2.00	0.046	-.0049817	-.0000483
mmarried# c.mage Married	-.0787984	.0175508	-4.49	0.000	-.1131973	-.0443996
prenatal1# c.mage Yes	-.0286228	.0176391	-1.62	0.105	-.0631948	.0059492
_cons	-2.928851	.8409119	-3.48	0.000	-4.577008	-1.280694

We specified option `aequations` to see the parameter estimates for the coefficients in the propensity-score model. There are eight coefficients, five on the base covariates (`mmarried`, `mage`, `fbaby`, `prenatal1`, and `_cons`) and three on the power-and-interaction covariates (`c.mage#c.mage`, `c.mage#1.mmarried`, and `c.mage#1.prenatal1`). Below we test whether the model balances all eight covariates.

```
. tebalance overid
Iteration 0: Criterion = .0602349
Iteration 1: Criterion = .06172749 (backed up)
Iteration 2: Criterion = .06428188 (backed up)
Iteration 3: Criterion = .06489132 (backed up)
Iteration 4: Criterion = .06527353 (backed up)
Iteration 5: Criterion = .0664376
Iteration 6: Criterion = .07184528
Iteration 7: Criterion = .0762921
Iteration 8: Criterion = .0767141
Iteration 9: Criterion = .07679677
Iteration 10: Criterion = .07699122
Iteration 11: Criterion = .0776527
Iteration 12: Criterion = .07771774
Iteration 13: Criterion = .07772609
Iteration 14: Criterion = .07773294
Iteration 15: Criterion = .077752
Iteration 16: Criterion = .07775324
Iteration 17: Criterion = .07775325
Iteration 18: Criterion = .07775325

Overidentification test for covariate balance
H0: Covariates are balanced

      chi2(8)      = 11.8612
Prob > chi2      = 0.1575
```

We cannot reject the null hypothesis that the IPW model balanced all eight covariates.

Below we specify option `bconly` to test whether the IPW model balanced the five base covariates only.

```
. tebalance overid, bconly
Iteration 0: Criterion = .1079977
Iteration 1: Criterion = .10800825 (backed up)
Iteration 2: Criterion = .10844177 (backed up)
Iteration 3: Criterion = .10851215 (backed up)
Iteration 4: Criterion = .10860849 (backed up)
Iteration 5: Criterion = .10907447
Iteration 6: Criterion = .110098
Iteration 7: Criterion = .11163978
Iteration 8: Criterion = .11260697
Iteration 9: Criterion = .11286395
Iteration 10: Criterion = .11331486
Iteration 11: Criterion = .11333976
Iteration 12: Criterion = .11335604
Iteration 13: Criterion = .11335696
Iteration 14: Criterion = .11335696

Overidentification test for covariate balance
H0: Covariates are balanced

      chi2(5)      = 7.82169
Prob > chi2      = 0.1663
```

We cannot reject the null hypothesis that the IPW model balanced the five base covariates.

Each test has a justification.

In a model-based approach, the [Imai and Ratkovic \(2014\)](#) test checks whether the propensity score is correctly specified. We include all eight covariates because they must all be balanced, if the propensity-score model is correctly specified.

A conditional-independence approach can be used to justify only including the base covariates in the test. In this approach, the propensity-score model need only balance the base covariates. Powers and interactions of the base covariates are included to get a propensity-score model that balances the base covariates, but balance of these higher-order terms is more than what needs to be checked.

In large samples, both tests should have nominal coverage under the null hypothesis that the propensity-score model is correctly specified. Under the alternative that the propensity-score model is misspecified, including all the covariates should yield a test with higher power.

The test that includes all the covariates is the default.



Stored results

`tebalance overid` stores the following in `r()`:

Scalars

<code>r(p)</code>	<i>p</i> -value
<code>r(df)</code>	overidentifying constraints, test degrees of freedom
<code>r(chi2)</code>	χ^2 statistic

References

- 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>.
- Imai, K., and M. Ratkovic. 2014. Covariate balancing propensity score. *Journal of the Royal Statistical Society, B ser.*, 76: 243–263. <https://doi.org/10.1111/rssb.12027>.

Also see

- [\[CAUSAL\] stteffects intro](#) — Introduction to treatment effects for observational survival-time data
- [\[CAUSAL\] tebalance](#) — Check balance after teffects or stteffects estimation
- [\[CAUSAL\] teffects aipw](#) — Augmented inverse-probability weighting
- [\[CAUSAL\] teffects ipw](#) — Inverse-probability weighting
- [\[CAUSAL\] teffects ipwra](#) — Inverse-probability-weighted regression adjustment
- [\[CAUSAL\] teoverlap](#) — Overlap plots

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

For suggested citations, see the FAQ on [citing Stata documentation](#).

