

**STATA TREATMENT-EFFECTS
REFERENCE MANUAL:
POTENTIAL OUTCOMES/COUNTERFACTUAL OUTCOMES
RELEASE 13**



A Stata Press Publication
StataCorp LP
College Station, Texas



Copyright © 1985–2013 StataCorp LP
All rights reserved
Version 13

Published by Stata Press, 4905 Lakeway Drive, College Station, Texas 77845
Typeset in \TeX

ISBN-10: 1-59718-128-5
ISBN-13: 978-1-59718-128-0

This manual is protected by copyright. All rights are reserved. No part of this manual may be reproduced, stored in a retrieval system, or transcribed, in any form or by any means—electronic, mechanical, photocopy, recording, or otherwise—without the prior written permission of StataCorp LP unless permitted subject to the terms and conditions of a license granted to you by StataCorp LP to use the software and documentation. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document.

StataCorp provides this manual “as is” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. StataCorp may make improvements and/or changes in the product(s) and the program(s) described in this manual at any time and without notice.

The software described in this manual is furnished under a license agreement or nondisclosure agreement. The software may be copied only in accordance with the terms of the agreement. It is against the law to copy the software onto DVD, CD, disk, diskette, tape, or any other medium for any purpose other than backup or archival purposes.

The automobile dataset appearing on the accompanying media is Copyright © 1979 by Consumers Union of U.S., Inc., Yonkers, NY 10703-1057 and is reproduced by permission from CONSUMER REPORTS, April 1979.

Stata, **STATA** Stata Press, Mata, **MATA** and NetCourse are registered trademarks of StataCorp LP.

Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations.

NetCourseNow is a trademark of StataCorp LP.

Other brand and product names are registered trademarks or trademarks of their respective companies.

For copyright information about the software, type `help copyright` within Stata.

The suggested citation for this software is

StataCorp. 2013. *Stata: Release 13*. Statistical Software. College Station, TX: StataCorp LP.

Contents

treatment effects	Introduction to treatment-effects commands	1
etpoisson	Poisson regression with endogenous treatment effects	2
etpoisson postestimation	Postestimation tools for etpoisson	14
etregress	Linear regression with endogenous treatment effects	17
etregress postestimation	Postestimation tools for etregress	31
teffects	Treatment-effects estimation for observational data	34
teffects intro	Introduction to treatment effects for observational data	35
teffects intro advanced	Advanced introduction to treatment effects for observational data	47
teffects aipw	Augmented inverse-probability weighting	60
teffects ipw	Inverse-probability weighting	81
teffects ipwra	Inverse-probability-weighted regression adjustment	89
teffects multivalued	Multivalued treatment effects	98
teffects nmatch	Nearest-neighbor matching	107
teffects overlap	Overlap plots	120
teffects postestimation	Postestimation tools for teffects	127
teffects psmatch	Propensity-score matching	134
teffects ra	Regression adjustment	142
Glossary		153
Subject and author index		157

Cross-referencing the documentation

When reading this manual, you will find references to other Stata manuals. For example,

[U] [26 Overview of Stata estimation commands](#)

[R] [regress](#)

[D] [reshape](#)

The first example is a reference to chapter 26, *Overview of Stata estimation commands*, in the *User's Guide*; the second is a reference to the `regress` entry in the *Base Reference Manual*; and the third is a reference to the `reshape` entry in the *Data Management Reference Manual*.

All the manuals in the Stata Documentation have a shorthand notation:

- [GSM] [Getting Started with Stata for Mac](#)
- [GSU] [Getting Started with Stata for Unix](#)
- [GSW] [Getting Started with Stata for Windows](#)
- [U] [Stata User's Guide](#)
- [R] [Stata Base Reference Manual](#)
- [D] [Stata Data Management Reference Manual](#)
- [G] [Stata Graphics Reference Manual](#)
- [XT] [Stata Longitudinal-Data/Panel-Data Reference Manual](#)
- [ME] [Stata Multilevel Mixed-Effects Reference Manual](#)
- [MI] [Stata Multiple-Imputation Reference Manual](#)
- [MV] [Stata Multivariate Statistics Reference Manual](#)
- [PSS] [Stata Power and Sample-Size Reference Manual](#)
- [P] [Stata Programming Reference Manual](#)
- [SEM] [Stata Structural Equation Modeling Reference Manual](#)
- [SVY] [Stata Survey Data Reference Manual](#)
- [ST] [Stata Survival Analysis and Epidemiological Tables Reference Manual](#)
- [TS] [Stata Time-Series Reference Manual](#)
- [TE] [Stata Treatment-Effects Reference Manual:
Potential Outcomes/Counterfactual Outcomes](#)
- [I] [Stata Glossary and Index](#)
- [M] [Mata Reference Manual](#)

Title

treatment effects — Introduction to treatment-effects commands

[Description](#) [Also see](#)

Description

This manual documents commands that use observational data to estimate the effect caused by getting one treatment instead of another. In observational data, treatment assignment is not controlled by those who collect the data; thus some common variables affect treatment assignment and treatment-specific outcomes. Observational data is sometimes called retrospective data or nonexperimental data, but to avoid confusion, we will always use the term “observational data”.

When all the variables that affect both treatment assignment and outcomes are observable, the outcomes are said to be conditionally independent of the treatment, and the `teffects` estimators may be used.

When not all of these variables common to both treatment assignment and outcomes are observable, the outcomes are not conditionally independent of the treatment, and `etregress` or `etpoisson` may be used.

`teffects` offers much flexibility in estimators and functional forms for the outcome models and the treatment-assignment models; see [\[TE\] teffects intro](#) or [\[TE\] teffects intro advanced](#).

`etregress` and `etpoisson` offer less flexibility than `teffects` because more structure must be imposed when conditional independence is not assumed. `etregress` is for linear outcomes and uses a normal distribution to model treatment assignment; see [\[TE\] etregress](#). `etpoisson` is for count outcomes and uses a normal distribution to model treatment assignment; see [\[TE\] etpoisson](#).

Endogenous treatment effects

[TE] etpoisson	Poisson regression with endogenous treatment effects
[TE] etregress	Linear regression with endogenous treatment effects

Treatment effects

[TE] teffects aipw	Augmented inverse-probability weighting
[TE] teffects ipw	Inverse-probability weighting
[TE] teffects ipwra	Inverse-probability-weighted regression adjustment
[TE] teffects nnmatch	Nearest-neighbor matching
[TE] teffects psmatch	Propensity-score matching
[TE] teffects ra	Regression adjustment

Also see

[\[U\] 1.3 What's new](#)
[\[TE\] teffects intro](#) — Introduction to treatment effects for observational data
[\[TE\] teffects intro advanced](#) — Advanced introduction to treatment effects for observational data
[\[TE\] teffects multivalued](#) — Multivalued treatment effects

Title

etpoisson — Poisson regression with endogenous treatment effects

Syntax	Menu	Description	Options
Remarks and examples	Stored results	Methods and formulas	Reference
Also see			

Syntax

```
etpoisson depvar [indepvars] [if] [in] [weight],  
    treat(depvart = indepvarst [, noconstant offset(varnameo)] ) [options]
```

<i>options</i>	Description
----------------	-------------

Model

* <u>treat</u> ()	equation for treatment effects
<u>noconstant</u>	suppress constant term
<u>exposure</u> (<i>varname</i> _{<i>e</i>})	include ln(<i>varname</i> _{<i>e</i>}) in model with coefficient constrained to 1
<u>offset</u> (<i>varname</i> _{<i>o</i>})	include <i>varname</i> _{<i>o</i>} in model with coefficient constrained to 1
<u>constraints</u> (<i>constraints</i>)	apply specified linear constraints
<u>collinear</u>	keep collinear variables

SE/Robust

<u>vce</u> (<i>vcetype</i>)	<i>vcetype</i> may be oim, <u>robust</u> , <u>cluster</u> <i>clustvar</i> , opg, <u>bootstrap</u> , or <u>jackknife</u>
-------------------------------	---

Reporting

<u>level</u> (#)	set confidence level; default is level(95)
<u>irr</u>	report incidence-rate ratios
<u>nocnsreport</u>	do not display constraints
<u>display_options</u>	control column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

Integration

<u>intpoints</u> (#)	use # Gauss–Hermite quadrature points; default is intpoints(24)
----------------------	---

Maximization

<u>maximize_options</u>	control the maximization process; seldom used
<u>coeflegend</u>	display legend instead of statistics

*treat() is required.

The full specification is treat(*depvar*_{*t*} = *indepvars*_{*t*} [, noconstant offset(*varname*_{*o*})]).

indepvars and *indepvars_t* may contain factor variables; see [U] 11.4.3 [Factor variables](#).

depvar, *depvar_t*, *indepvars*, and *indepvars_t* may contain time-series operators; see [U] 11.4.4 [Time-series varlists](#).

bootstrap, *by*, *jackknife*, *rolling*, *statsby*, and *svy* are allowed; see [U] 11.1.10 [Prefix commands](#).

Weights are not allowed with the *bootstrap* prefix; see [R] [bootstrap](#).

Weights are not allowed with the *jackknife* prefix; see [R] [jackknife](#).

vce() and weights are not allowed with the *svy* prefix; see [SVY] [svy](#).

fweights, *aweight*, *iweight*, and *pweight* are allowed; see [U] 11.1.6 [weight](#).

coeflegend does not appear in the dialog box.

See [U] 20 [Estimation and postestimation commands](#) for more capabilities of estimation commands.

Menu

Statistics > Treatment effects > Endogenous treatment estimators > Count outcome

Description

etpoisson estimates the parameters of a Poisson regression model in which one of the regressors is an endogenous binary treatment. Both the average treatment effect and the average treatment effect on the treated can be estimated with *etpoisson*.

Options

Model

treat(*depvar_t* = *indepvars_t* [, *noconstant* *offset*(*varname_o*)]) specifies the variables and options for the treatment equation. It is an integral part of specifying a treatment-effects model and is required.

The indicator of treatment, *depvar_t*, should be coded as 0 or 1.

noconstant, *exposure*(*varname_e*), *offset*(*varname_o*), *constraints*(*constraints*), *collinear*; see [R] [estimation options](#).

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (*oim*, *opg*), that are robust to some kinds of misspecification (*robust*), that allow for intragroup correlation (*cluster* *clustvar*), and that use bootstrap or jackknife methods (*bootstrap*, *jackknife*); see [R] [vce_option](#).

Reporting

level(#); see [R] [estimation options](#).

irr reports estimated coefficients transformed to incidence-rate ratios, that is, e^{β_i} rather than β_i . Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated or stored. *irr* may be specified at estimation or when replaying previously estimated results.

nocnsreport; see [R] [estimation options](#).

display_options: *noomitted*, *vsquish*, *noemptycells*, *baselevels*, *allbaselevels*, *nofvlabel*, *fvwrap*(#), *fvwrapon*(*style*), *cformat*(*%fmt*), *pformat*(*%fmt*), *sformat*(*%fmt*), and *no1stretch*; see [R] [estimation options](#).

Integration

`intpoints(#)` specifies the number of integration points to use for integration by quadrature. The default is `intpoints(24)`; the maximum is `intpoints(128)`. Increasing this value improves the accuracy but also increases computation time. Computation time is roughly proportional to its value.

Maximization

`maximize_options`: `difficult`, `technique(algorithm_spec)`, `iterate(#)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, `nonrtolerance`, and `from(init_specs)`; see [R] [maximize](#). These options are seldom used.

Setting the optimization type to `technique(bhhh)` resets the default `vcetype` to `vce(opg)`.

The following option is available with `etpoisson` but is not shown in the dialog box:

`coeflegend`; see [R] [estimation options](#).

Remarks and examples

Remarks are presented under the following headings:

[Overview](#)

[Basic example](#)

[Average treatment effect \(ATE\)](#)

[Average treatment effect on the treated \(ATET\)](#)

Overview

`etpoisson` estimates the parameters of a Poisson regression model that includes an endogenous binary-treatment variable. The dependent variable must be a Poisson distributed count. The parameters estimated by `etpoisson` can be used to estimate the average treatment effect (ATE) and average treatment effect on the treated (ATET).

We call the model fit by `etpoisson` an endogenous treatment-regression model, although it is also known as an endogenous binary-variable model or as an endogenous dummy-variable model. The endogenous treatment-regression model fit by `etpoisson` is a specific endogenous treatment-effects model; it uses a nonlinear model for the outcome and a constrained normal distribution to model the deviation from the conditional independence assumption imposed by the estimators implemented by `teffects`; see [TE] [teffects intro](#). In treatment-effects jargon, the endogenous binary-variable model fit by `etpoisson` is a nonlinear potential-outcome model that allows for a specific correlation structure between the unobservables that affect the treatment and the unobservables that affect the potential outcomes. See [TE] [etregress](#) for an estimator that allows for a linear-outcome model and a similar model for the endogeneity of the treatment.

More formally, we have an equation for outcome y_j and an equation for treatment t_j :

$$E(y_j | \mathbf{x}_j, t_j, \epsilon_j) = \exp(\mathbf{x}_j \boldsymbol{\beta} + \delta t_j + \epsilon_j)$$

$$t_j = \begin{cases} 1, & \mathbf{w}_j \boldsymbol{\gamma} + u_j > 0 \\ 0, & \text{otherwise} \end{cases}$$

The \mathbf{x}_j are the covariates used to model the outcome, \mathbf{w}_j are the covariates used to model treatment assignment, and error terms ϵ_j and u_j are bivariate normal with mean 0 and covariance matrix

$$\begin{bmatrix} \sigma^2 & \sigma\rho \\ \sigma\rho & 1 \end{bmatrix}$$

The covariates \mathbf{x}_j and \mathbf{w}_j are unrelated to the error terms; in other words, they are exogenous. Note that y_j may be a count or continuous and nonnegative in this specification.

Terza (1998) describes the maximum likelihood estimator used in `etpoisson`. Terza (1998) categorized the model fit by `etpoisson` as an endogenous-switching model. These models involve a binary switch that is endogenous for the outcome. Calculation of the maximum likelihood estimate involves numeric approximation of integrals via Gauss–Hermite quadrature. This is computationally intensive, but the computational costs are reasonable on modern computers.

Basic example

▷ Example 1

In this example, we observe a simulated random sample of 5,000 households. The outcome of interest is the number of trips taken by members of the household in the 24-hour period immediately prior to the interview time.

We have fictional household level data on the following variables: number of trips taken in the past 24 hours (`trips`), distance to the central business district from the household (`cbd`), distance from the household to a public transit node (`ptn`), an indicator of whether there is a full-time worker in the household (`worker`), an indicator of whether the examined period is on a weekend (`weekend`), the ratio of the household income to the median income of the census tract (`realinc`), and an indicator of car ownership (`owncar`). We suspect that unobservables that affect the number of trips also affect the household’s propensity to own a car.

We use `etpoisson` to estimate the parameters of a Poisson regression model for the number of trips with car ownership as an endogenous treatment. In subsequent examples, we will use `margins` (see [R] [margins](#)) to estimate the ATE and the ATET of car ownership on the number of trips taken by the household. In the `etpoisson` command below, we specify the `vce(robust)` option because we need to specify `vce(unconditional)` when we use `margins` later.

```

. use http://www.stata-press.com/data/r13/trip1
(Household trips, car ownership)

. etpoisson trips cbd ptn worker weekend,
> treat(owncar = cbd ptn worker realinc) vce(robust)

Iteration 0:  log pseudolikelihood = -14845.147 (not concave)
Iteration 1:  log pseudolikelihood = -14562.997 (not concave)
Iteration 2:  log pseudolikelihood = -13655.592 (not concave)
Iteration 3:  log pseudolikelihood = -12847.219 (not concave)
Iteration 4:  log pseudolikelihood = -12566.037
Iteration 5:  log pseudolikelihood = -12440.974
Iteration 6:  log pseudolikelihood = -12413.485
Iteration 7:  log pseudolikelihood = -12412.699
Iteration 8:  log pseudolikelihood = -12412.696
Iteration 9:  log pseudolikelihood = -12412.696

Poisson regression with endogenous treatment      Number of obs   =       5000
(24 quadrature points)                          Wald chi2(5)    =       397.94
Log pseudolikelihood = -12412.696                Prob > chi2     =       0.0000

```

	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
trips						
cbd	-.0100919	.0020071	-5.03	0.000	-.0140258	-.006158
ptn	-.0204038	.0020289	-10.06	0.000	-.0243805	-.0164272
worker	.692301	.0548559	12.62	0.000	.5847854	.7998166
weekend	.0930517	.034538	2.69	0.007	.0253585	.160745
1.owncar	.5264713	.1124157	4.68	0.000	.3061406	.746802
_cons	-.2340772	.0810812	-2.89	0.004	-.3929934	-.0751609
owncar						
cbd	.007218	.00239	3.02	0.003	.0025337	.0119023
ptn	.0084769	.0024518	3.46	0.001	.0036714	.0132824
worker	.543643	.0504267	10.78	0.000	.4448085	.6424774
realinc	.176479	.0108746	16.23	0.000	.1551652	.1977928
_cons	-.4611246	.0592161	-7.79	0.000	-.5771859	-.3450633
/athrho	.5741169	.0957832	5.99	0.000	.3863852	.7618486
/lnsigma	-.2182037	.0256281	-8.51	0.000	-.2684338	-.1679735
rho	.5183763	.0700449			.3682398	.6421645
sigma	.8039617	.020604			.764576	.8453762

Wald test of indep. eqns. (rho = 0): chi2(1) = 35.93 Prob > chi2 = 0.0000

The Wald test in the header is highly significant, indicating a good model fit. All the covariates are statistically significant, and the Wald test in the footer indicates that we can reject the null hypothesis of no correlation between the treatment errors and the outcome errors.

We can interpret the coefficient on 1.owncar as the logarithm of the ratio of the treatment potential-outcome mean to the control potential-outcome mean. The treatment variable did not interact with any of the outcome covariates, so the effect of each regressor is the same in the two regimes and will cancel from the ratio of potential-outcome means. This means the ratio is equivalent to the exponentiated coefficient on 1.owncar. After discussing the other parameters, we will use `lincom` to obtain this ratio. See [R] [lincom](#) for more information.

The estimated correlation between the treatment-assignment errors and the outcome errors is 0.518, indicating that unobservables that increase the number of trips tend to occur with unobservables that increase the chance of car ownership.

The results for the two ancillary parameters require explanation. `etpoisson` estimates the inverse hyperbolic tangent of ρ ,

$$\operatorname{atanh} \rho = \frac{1}{2} \ln \left(\frac{1 + \rho}{1 - \rho} \right)$$

and $\ln \sigma$ rather than ρ and σ . For numerical stability during optimization, `etpoisson` does not directly estimate ρ and σ .

Now we use `lincom` and the `eform` option to estimate the exponentiated coefficient for `1.owncar`. This corresponds to the ratio of the treatment regime potential-outcome mean to the control regime potential-outcome mean.

```
. lincom [trips]_b[1.owncar], eform
(1) [trips]1.owncar = 0
```

	exp(b)	Std. Err.	z	P> z	[95% Conf. Interval]	
(1)	1.692948	.1903139	4.68	0.000	1.358173	2.110241

The potential-outcome mean for the treatment regime is 1.69 times the potential-outcome mean for the control regime. So the average number of trips in the treatment regime is over one and a half times the average number of trips in the control regime.

By interacting the treatment, `owncar`, with the other regressors, we could estimate different coefficients for the regressors in the treatment and control regimes. In the current model, there are no treatment interactions, so the coefficients are the same in each regime.



Average treatment effect (ATE)

The parameter estimates from `etpoisson` can be used by `margins` to estimate the ATE, the average difference of the treatment and control potential outcomes.

► Example 2

Continuing with [example 1](#), we use `margins` to estimate the ATE of car ownership on the number of trips taken in a 24-hour period.

We can estimate the ATE of car ownership by using the potential-outcome means obtained through the `predict`, `pomean` command and the `margins` command; see [Methods and formulas](#) below and [\[TE\] etpoisson postestimation](#) for more details about the use of `predict` after `etpoisson`.

The `r.` notation indicates that the potential-outcome means for treatment and control will be contrasted. We specify the `contrast(nowald)` option to suppress the Wald tests that `margins` displays by default for contrasts.

```
. margins r.owncar, vce(unconditional) contrast(nowald)
Contrasts of predictive margins
Expression : Potential-outcome mean, predict()
```

	Unconditional			
	Contrast	Std. Err.	z	[95% Conf. Interval]
owncar (1 vs 0)	1.058914	.1922909	.6820309	1.435797

The estimated ATE of car ownership on the number of trips taken is 1.06. The average household will take 1.06 more trips when it owns a car.

◀

Average treatment effect on the treated (ATET)

The parameter estimates from `etpoisson` can be used by `margins` to estimate the ATET, the average difference of the treatment and control potential outcomes in the treated population.

▶ Example 3

Continuing with the [previous example](#), we use `margins` to estimate the ATET of car ownership on the number of trips taken in a 24-hour period.

We can estimate the ATET of car ownership by using the conditional treatment effect (conditional on exogenous covariates and treatment level) obtained through the `predict`, `cte` command and the `margins` command; see [Methods and formulas](#) below and [\[TE\] etpoisson postestimation](#) for more details about the use of `predict` after `etpoisson`.

We estimate the ATET with `margins`. We specify `cte` in the `predict()` option. Estimation is restricted to the treated subpopulation by specifying `owncar` in the `subpop()` option.

```
. margins, predict(cte) vce(unconditional) subpop(owncar)
Predictive margins          Number of obs =          5000
                          Subpop. no. obs =          3504
Expression   : Conditional treatment effect, predict(cte)
```

	Unconditional		z	P> z	[95% Conf. Interval]	
	Margin	Std. Err.				
_cons	1.251971	.2059201	6.08	0.000	.8483747	1.655567

The estimated ATET of car ownership on the number of trips taken is 1.25. Thus the average household in the treated population will take 1.25 more trips than it would if it did not own a car. This number is higher than the ATE. In this model, the ATE and ATET will only coincide when there is no correlation between the treatment errors and outcome errors and the exogenous covariates x have the same distribution in the general population and treated subpopulation. See [Methods and formulas](#) for more details.

◀

Stored results

etpoisson stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(k)</code>	number of parameters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_eq_model)</code>	number of equations in overall model test
<code>e(k_aux)</code>	number of auxiliary parameters
<code>e(k_dv)</code>	number of dependent variables
<code>e(df_m)</code>	model degrees of freedom
<code>e(ll)</code>	log likelihood
<code>e(N_clust)</code>	number of clusters
<code>e(chi2)</code>	χ^2
<code>e(chi2_c)</code>	χ^2 for comparison, $\rho=0$ test
<code>e(n_quad)</code>	number of quadrature points
<code>e(p)</code>	significance
<code>e(p_c)</code>	significance of comparison test
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(ic)</code>	number of iterations
<code>e(rc)</code>	return code
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	etpoisson
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(title2)</code>	secondary title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(offset1)</code>	offset for regression equation
<code>e(offset2)</code>	offset for treatment equation
<code>e(chi2type)</code>	Wald; type of model χ^2 test
<code>e(chi2_ct)</code>	Wald; type of comparison χ^2 test
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. Err.
<code>e(opt)</code>	type of optimization
<code>e(which)</code>	max or min; whether optimizer is to perform maximization or minimization
<code>e(ml_method)</code>	type of ml method
<code>e(user)</code>	name of likelihood-evaluator program
<code>e(technique)</code>	maximization technique
<code>e(properties)</code>	<code>b V</code>
<code>e(predict)</code>	program used to implement predict
<code>e(marginsok)</code>	predictions allowed by margins
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(Cns)</code>	constraints matrix
<code>e(ilog)</code>	iteration log (up to 20 iterations)
<code>e(gradient)</code>	gradient vector
<code>e(V)</code>	variance-covariance matrix of the estimators
<code>e(V_modelbased)</code>	model-based variance

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

Methods and formulas

Terza (1998) derives the maximum likelihood estimator implemented here. We provide some details of the derivation and then explain how the model is nested in a more general potential-outcomes model. Then the ATE and ATET are derived.

Let \mathbf{x}_j be the covariates used to model the outcome, and let \mathbf{w}_j be the covariates used to model treatment assignment. Define $\mathbf{z}_j = (\mathbf{w}_j, \mathbf{x}_j)$. The vector \mathbf{z}_j contains all the exogenous covariates in the model. When offsets o_j^β are used in the outcome variable equation, the following formulas apply with $\mathbf{x}_j\beta$ changed to $\mathbf{x}_j\beta + o_j^\beta$. Similarly, when offsets o_j^γ are used in the endogenous treatment equation, the following formulas apply with $\mathbf{w}_j\gamma$ changed to $\mathbf{w}_j\gamma + o_j^\gamma$. If offsets are used in either equation, they are included in the vector of exogenous covariates \mathbf{z}_j .

For treatment t_j , \mathbf{z}_j , and ϵ_j , outcome y_j of this model has conditional mean

$$E(y_j|\mathbf{x}_j, t_j, \epsilon_j) = \exp(\mathbf{x}_j\beta + \delta t_j + \epsilon_j) \quad (1)$$

The probability density function of y_j for this model, conditioned on treatment t_j , \mathbf{z}_j , and ϵ_j , is given by

$$f(y_j|\mathbf{z}_j, t_j, \epsilon_j) = \frac{\exp\{-\exp(\mathbf{x}_j\beta + \delta t_j + \epsilon_j)\} \{\exp(\mathbf{x}_j\beta + \delta t_j + \epsilon_j)\}^{y_j}}{y_j!}$$

The treatment t_j is determined by

$$t_j = \begin{cases} 1, & \text{if } \mathbf{w}_j\gamma + u_j > 0 \\ 0, & \text{otherwise} \end{cases}$$

The error terms ϵ_j and u_j are bivariate normal with mean zero and covariance matrix

$$\begin{bmatrix} \sigma^2 & \sigma\rho \\ \sigma\rho & 1 \end{bmatrix}$$

Conditional on ϵ_j , u_j is normal with mean $\epsilon_j\rho/\sigma$ and variance $(1 - \rho^2)$; thus we obtain the following conditional probability density for t_j :

$$\Pr(t_j|\mathbf{z}_j, \epsilon_j) = t_j \Phi \left\{ \frac{\mathbf{w}_j\gamma + (\rho/\sigma)\epsilon_j}{\sqrt{1 - \rho^2}} \right\} + (1 - t_j) \left[1 - \Phi \left\{ \frac{\mathbf{w}_j\gamma + (\rho/\sigma)\epsilon_j}{\sqrt{1 - \rho^2}} \right\} \right]$$

Φ denotes the standard normal cumulative distribution function. This leads to the following joint density of y_j , t_j , and ϵ_j :

$$f(y_j, t_j, \epsilon_j|\mathbf{z}_j) = f(y_j|\mathbf{z}_j, t_j, \epsilon_j)P(t_j|\mathbf{z}_j, \epsilon_j)f(\epsilon_j)$$

The density of y_j and t_j , conditioned on \mathbf{z}_j , is obtained by integrating the above with respect to ϵ_j . Recall that ϵ_j is normal with mean 0 and variance σ^2 .

$$f(y_j, t_j|\mathbf{z}_j) = \int_{-\infty}^{\infty} f(y_j|\mathbf{z}_j, t_j, \epsilon_j)P(t_j|\mathbf{z}_j, \epsilon_j) \frac{1}{\sigma\sqrt{2\pi}} \exp \left\{ - \left(\frac{\epsilon_j}{\sigma\sqrt{2}} \right)^2 \right\} d\epsilon_j$$

$f(y_j, t_j | \mathbf{z}_j)$ cannot be evaluated in a closed form. We change the variable of integration from ϵ_j to $\eta_j = \epsilon_j / (\sigma\sqrt{2})$, which yields

$$f(y_j, t_j | \mathbf{z}_j) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} f(y_j | \mathbf{z}_j, t_j, \sqrt{2}\sigma\eta_j) P(t_j | \mathbf{z}_j, \sqrt{2}\sigma\eta_j) \exp(-\eta_j^2) d\eta_j$$

We approximate this integral by Gauss–Hermite quadrature. Observing a sample of t_j , y_j , and \mathbf{z}_j , we calculate the log likelihood as the following:

$$\ln L = \sum_{j=1}^n w_j \ln \{f(y_j, t_j | \mathbf{z}_j)\}$$

The w_j terms denote optional weights.

In the maximum likelihood estimation, σ and ρ are not directly estimated. Directly estimated are $\ln \sigma$ and $\operatorname{atanh} \rho$:

$$\operatorname{atanh} \rho = \frac{1}{2} \ln \left(\frac{1 + \rho}{1 - \rho} \right)$$

Now we present formulas for the ATE and ATET. First, we nest the endogenous-treatment Poisson regression model in a potential-outcome model. A potential-outcome model specifies what each individual would obtain in each treatment level.

A potential-outcome model that nests the endogenous-treatment Poisson regression fit by `etpoisson` is

$$\begin{aligned} E(y_{0j} | \mathbf{x}_j, \epsilon_j) &= \exp(\mathbf{x}_j \boldsymbol{\beta}_0 + \epsilon_{0j}) \\ E(y_{1j} | \mathbf{x}_j, \epsilon_j) &= \exp(\mathbf{x}_j \boldsymbol{\beta}_1 + \epsilon_{1j}) \\ t_j &= \begin{cases} 1, & \text{if } \mathbf{w}_j \boldsymbol{\gamma} + u_j > 0 \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

where y_{0j} is the outcome that person j obtains if person j selects treatment 0, and y_{1j} is the outcome that person j obtains if person j selects treatment 1. This formulation allows differing coefficients for the control ($\boldsymbol{\beta}_0$) and treatment ($\boldsymbol{\beta}_1$) regimes. The constant intercept for the control group is β_{00} . The constant intercept for the treatment group is $\beta_{11} = \beta_{00} + \delta$, where δ is the coefficient for treatment t_j in the outcome (1). The remaining notation was defined above.

We may allow other coefficients to differ across regimes in the outcome (1) by adding interactions between the treatment t_j and covariates \mathbf{x}_j to the model. To be concise, we use two coefficient vectors $\boldsymbol{\beta}_0$ and $\boldsymbol{\beta}_1$ here rather than a single coefficient vector with interactions between the treatment t_j and covariates \mathbf{x}_j . The two formulations are equivalent.

We never observe both y_{0j} and y_{1j} , only one or the other. We observe

$$y_j = t_j y_{1j} + (1 - t_j) y_{0j}$$

The vector of error terms $(\epsilon_{0j}, \epsilon_{1j}, u_j)'$ comes from a mean zero trivariate normal distribution with covariance matrix

$$\begin{bmatrix} \sigma^2 & \theta & \sigma\rho \\ \theta & \sigma^2 & \sigma\rho \\ \sigma\rho & \sigma\rho & 1 \end{bmatrix}$$

The parameters σ and ρ were discussed earlier. The parameter θ is the covariance between the two potential outcomes. We cannot estimate θ because we have no observations in which an individual is observed in both potential outcomes. Fortunately, θ is not required for the calculations that we present.

The ATE is the difference in means of the potential outcomes. The mean of each potential outcome accounts for each individual's contribution, regardless of whether that individual selects that treatment level.

The conditional means of the potential outcomes y_{tj} , $t \in (0, 1)$ for exogenous covariates \mathbf{z}_j are

$$E(y_{tj}|\mathbf{z}_j) = \exp\left(\mathbf{x}_j\beta_t + \frac{\sigma^2}{2}\right)$$

We can see that when the coefficients are the same across the regimes, the ratio of potential-outcome means will be equal to $\exp(\delta)$; this is true of the conditional and marginal potential-outcome means.

The difference in potential-outcome means or treatment effect at exogenous covariates \mathbf{z}_j is

$$E(y_{1j} - y_{0j}|\mathbf{z}_j) = \{\exp(\mathbf{x}_j\beta_1) - \exp(\mathbf{x}_j\beta_0)\} \exp\left(\frac{\sigma^2}{2}\right)$$

By the law of iterated expectations, the ATE is

$$\begin{aligned} E(y_{1j} - y_{0j}) &= E\{E(y_{1j} - y_{0j}|\mathbf{z}_j)\} \\ &= E\left[\{\exp(\mathbf{x}_j\beta_1) - \exp(\mathbf{x}_j\beta_0)\} \exp\left(\frac{\sigma^2}{2}\right)\right] \end{aligned}$$

This expectation can be estimated as a predictive margin.

Now we will derive an expression for the ATET.

The conditional means of the potential outcomes y_{tj} , $t \in (0, 1)$ for exogenous covariates \mathbf{z}_j and treatment t_j are

$$E(y_{tj}|\mathbf{z}_j, t_j) = \exp\left(\mathbf{x}_j\beta_t + \frac{\sigma^2}{2}\right) \left\{ \frac{\Phi(\rho\sigma + \mathbf{w}_j\gamma)}{\Phi(\mathbf{w}_j\gamma)} \right\}^{t_j} \left\{ \frac{1 - \Phi(\rho\sigma + \mathbf{w}_j\gamma)}{1 - \Phi(\mathbf{w}_j\gamma)} \right\}^{1-t_j}$$

Rather than the conditional potential-outcome means, the conditional mean of the observed outcome may be of interest. The conditional mean of the observed outcome y_j for endogenous treatment indicator t_j and exogenous covariates \mathbf{z}_j is given by

$$\begin{aligned}
 E(y_j | \mathbf{z}_j, t_j) &= t_j \exp\left(\mathbf{x}_j \boldsymbol{\beta}_1 + \frac{\sigma^2}{2}\right) \frac{\Phi(\rho\sigma + \mathbf{w}_j \boldsymbol{\gamma})}{\Phi(\mathbf{w}_j \boldsymbol{\gamma})} \\
 &+ (1 - t_j) \exp\left(\mathbf{x}_j \boldsymbol{\beta}_0 + \frac{\sigma^2}{2}\right) \frac{1 - \Phi(\rho\sigma + \mathbf{w}_j \boldsymbol{\gamma})}{1 - \Phi(\mathbf{w}_j \boldsymbol{\gamma})}
 \end{aligned}$$

The treatment effect at exogenous covariates \mathbf{z}_j and treatment t_j is

$$\begin{aligned}
 E(y_{1j} - y_{0j} | \mathbf{z}_j, t_j) &= \\
 \{ \exp(\mathbf{x}_j \boldsymbol{\beta}_1) - \exp(\mathbf{x}_j \boldsymbol{\beta}_0) \} \exp\left(\frac{\sigma^2}{2}\right) &\left\{ \frac{\Phi(\rho\sigma + \mathbf{w}'_j \boldsymbol{\gamma})}{\Phi(\mathbf{w}_j \boldsymbol{\gamma})} \right\}^{t_j} \left\{ \frac{1 - \Phi(\rho\sigma + \mathbf{w}_j \boldsymbol{\gamma})}{1 - \Phi(\mathbf{w}'_j \boldsymbol{\gamma})} \right\}^{1-t_j}
 \end{aligned}$$

By the law of iterated expectations, the ATET is

$$\begin{aligned}
 E(y_{1j} - y_{0j} | t_j = 1) &= E\{E(y_{1j} - y_{0j} | \mathbf{z}_j, t_j = 1) | t_j = 1\} \\
 &= E\left[\{ \exp(\mathbf{x}_j \boldsymbol{\beta}_1) - \exp(\mathbf{x}_j \boldsymbol{\beta}_0) \} \exp\left(\frac{\sigma^2}{2}\right) \frac{\Phi(\rho\sigma + \mathbf{w}_j \boldsymbol{\gamma})}{\Phi(\mathbf{w}_j \boldsymbol{\gamma})} \Big| t_j = 1\right]
 \end{aligned}$$

This can be estimated as a predictive margin on the treated subpopulation.

We note that when $\rho = 0$, the correction factor involving Φ will disappear from the ATET. Then the ATE and ATET will be equivalent if the distribution of \mathbf{x}_j under the treated population is identical to the distribution over the entire population.

The probability of y_j conditional on t_j and \mathbf{z}_j is

$$\Pr(y_j = n | \mathbf{z}_j, t_j) = \frac{f(y_j = n, t_j | \mathbf{z}_j)}{\Phi(\mathbf{w}_j \boldsymbol{\gamma})^{t_j} \Phi(-\mathbf{w}_j \boldsymbol{\gamma})^{1-t_j}}$$

As discussed earlier, we approximate $f(y_j, t_j | \mathbf{z}_j)$ using Gauss–Hermite quadrature.

Reference

Terza, J. V. 1998. Estimating count data models with endogenous switching: Sample selection and endogenous treatment effects. *Journal of Econometrics* 84: 129–154.

Also see

- [TE] [etpoisson postestimation](#) — Postestimation tools for etpoisson
- [TE] [etregress](#) — Linear regression with endogenous treatment effects
- [R] [ivpoisson](#) — Poisson regression with endogenous regressors
- [R] [ivprobit](#) — Probit model with continuous endogenous regressors
- [R] [ivregress](#) — Single-equation instrumental-variables regression
- [R] [ivtobit](#) — Tobit model with continuous endogenous regressors
- [R] [poisson](#) — Poisson regression
- [SVY] [svy estimation](#) — Estimation commands for survey data
- [U] [20 Estimation and postestimation commands](#)

Title

etpoisson postestimation — Postestimation tools for etpoisson

Description Syntax for predict Menu for predict Options for predict
Remarks and examples Methods and formulas Also see

Description

The following standard postestimation commands are available after `etpoisson`:

Command	Description
<code>contrast</code>	contrasts and ANOVA-style joint tests of estimates
<code>estat ic</code>	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estat (svy)</code>	postestimation statistics for survey data
<code>estimates</code>	cataloging estimation results
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of coefficients
<code>lrtest</code> ¹	likelihood-ratio test
<code>margins</code>	marginal means, predictive margins, marginal effects, and average marginal effects
<code>marginsplot</code>	graph the results from margins (profile plots, interaction plots, etc.)
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
<code>predict</code>	predictions, probabilities, and treatment effects
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>pwcompare</code>	pairwise comparisons of estimates
<code>suest</code>	seemingly unrelated estimation
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

¹ `lrtest` is not appropriate with `svy` estimation results.

Syntax for predict

```
predict [type] newvar [if] [in] [, statistic nooffset]
```

```
predict [type] { stub* | newvarreg newvartreat newvarathrho newvarlnsigma }
      [if] [in] , scores
```

<i>statistic</i>	Description
------------------	-------------

Main

<u>pomean</u>	potential-outcome mean (the predicted count); the default
<u>omean</u>	observed-outcome mean (the predicted count)
<u>cte</u>	conditional treatment effect at treatment level
<u>pr(<i>n</i>)</u>	probability $\Pr(y_j = n)$
<u>pr(<i>a</i>,<i>b</i>)</u>	probability $\Pr(a \leq y_j \leq b)$
<u>xb</u>	linear prediction
<u>xbtreat</u>	linear prediction for treatment equation

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

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

pomean, the default, calculates the potential-outcome mean.

omean calculates the observed-outcome mean.

cte calculates the treatment effect, the difference of potential-outcome means, conditioned on treatment level.

pr(*n*) calculates the probability $\Pr(y_j = n)$, where *n* is a nonnegative integer that may be specified as a number or a variable.

pr(*a*,*b*) calculates the probability $\Pr(a \leq y_j \leq b)$, where *a* and *b* are nonnegative integers that may be specified as numbers or variables;

b missing ($b \geq .$) means $+\infty$;

pr(20, .) calculates $\Pr(y_j \geq 20)$;

pr(20,*b*) calculates $\Pr(y_j \geq 20)$ in observations for which $b \geq .$ and calculates $\Pr(20 \leq y_j \leq b)$ elsewhere.

pr(.,*b*) produces a syntax error. A missing value in an observation of the variable *a* causes a missing value in that observation for pr(*a*,*b*).

xb calculates the linear prediction for the dependent count variable, which is $\mathbf{x}_j\beta$ if neither `offset()` nor `exposure()` was specified; $\mathbf{x}_j\beta + \text{offset}_j^\beta$ if `offset()` was specified; or $\mathbf{x}_j\beta + \ln(\text{exposure}_j)$ if `exposure()` was specified.

`xbtreat` calculates the linear prediction for the endogenous treatment equation, which is $w_j\gamma$ if `offset()` was not specified in `treat()` and $w_j\gamma + \text{offset}_j^\alpha$ if `offset()` was specified in `treat()`. `nooffset` is relevant only if you specified `offset()` or `exposure()` when you fit the model. It modifies the calculations made by `predict` so that they ignore the offset or exposure variable. `nooffset` removes the offset from calculations involving both the `treat()` equation and the dependent count variable.

`scores` calculates equation-level score variables.

The first new variable will contain $\partial \ln L / \partial (\mathbf{x}_j \beta)$.

The second new variable will contain $\partial \ln L / \partial (w_j \gamma)$.

The third new variable will contain $\partial \ln L / \partial \text{atanh } \rho$.

The fourth new variable will contain $\partial \ln L / \partial \ln \sigma$.

Remarks and examples

The average treatment effect (ATE) and the average treatment effect on the treated (ATET) are the parameters most frequently estimated by postestimation techniques after [etpoisson](#).

You can use the `margins` command (see [\[R\] margins](#)) after [etpoisson](#) to estimate the ATE or ATET. See [example 2](#) of [\[TE\] etpoisson](#) for an example of ATE estimation. See [example 3](#) of [\[TE\] etpoisson](#) for an example of ATET estimation.

See [example 1](#) of [\[TE\] etpoisson](#) for an example using `lincom` after [etpoisson](#).

Methods and formulas

See [Methods and formulas](#) of [\[TE\] etpoisson](#) for details.

Also see

[\[TE\] etpoisson](#) — Poisson regression with endogenous treatment effects

[\[U\] 20 Estimation and postestimation commands](#)

Title

etregress — Linear regression with endogenous treatment effects

Syntax

Description

Options for two-step consistent estimates

Stored results

References

Menu

Options for maximum likelihood estimates

Remarks and examples

Methods and formulas

Also see

Syntax

Basic syntax

```
etregress depvar [indepvars], treat(depvart = indepvarst) [twostep]
```

Full syntax for maximum likelihood estimates only

```
etregress depvar [indepvars] [if] [in] [weight],  
treat(depvart = indepvarst [, noconstant]) [etregress_ml_options]
```

Full syntax for two-step consistent estimates only

```
etregress depvar [indepvars] [if] [in],  
treat(depvart = indepvarst [, noconstant]) twostep [etregress_ts_options]
```

etregress_ml_options

Description

Model

* <u>treat</u> ()	equation for treatment effects
<u>noconstant</u>	suppress constant term
<u>constraints</u> (<i>constraints</i>)	apply specified linear constraints
<u>collinear</u>	keep collinear variables

SE/Robust

<u>vce</u> (<i>vcetype</i>)	<i>vcetype</i> may be <u>oim</u> , <u>robust</u> , <u>cluster</u> <i>clustvar</i> , <u>opg</u> , <u>bootstrap</u> , or <u>jackknife</u>
-------------------------------	---

Reporting

<u>level</u> (#)	set confidence level; default is <u>level</u> (95)
<u>first</u>	report first-step probit estimates
<u>noskip</u>	perform likelihood-ratio test
<u>hazard</u> (<i>newvar</i>)	create <i>newvar</i> containing hazard from treatment equation
<u>nocnsreport</u>	do not display constraints
<u>display_options</u>	control column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

Maximization

<u>maximize_options</u>	control the maximization process; seldom used
<u>coeflegend</u>	display legend instead of statistics

*treat(*depvar*_{*t*} = *indepvars*_{*t*} [, noconstant]) is required.

<i>etregress</i> <i>_ts_</i> <i>options</i>	Description
Model	
* treat ()	equation for treatment effects
* twostep	produce two-step consistent estimate
noconstant	suppress constant term
SE	
vce (<i>vcetype</i>)	<i>vcetype</i> may be conventional, bootstrap , or jackknife
Reporting	
level (#)	set confidence level; default is level(95)
first	report first-step probit estimates
hazard (<i>newvar</i>)	create <i>newvar</i> containing hazard from treatment equation
display_options	control column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
coeflegend	display legend instead of statistics
* treat (<i>devar</i> _{<i>t</i>} = <i>indepvars</i> _{<i>t</i>} [, noconstant]) and twostep are required.	

indepvars and *indepvars*_{*t*} may contain factor variables; see [U] 11.4.3 Factor variables.

devar, *indepvars*, *devar*_{*t*}, and *indepvars*_{*t*} may contain time-series operators; see [U] 11.4.4 Time-series varlists.

bootstrap, **by**, **fp**, **jackknife**, **rolling**, **statsby**, and **svy** are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the **bootstrap** prefix; see [R] **bootstrap**.

aweights are not allowed with the **jackknife** prefix; see [R] **jackknife**.

twostep, **vce**() , **first**, **noskip**, **hazard**() , and weights are not allowed with the **svy** prefix; see [SVY] **svy**.

pweights, **aweight**s, **fweight**s, and **iweight**s are allowed with maximum likelihood estimation; see [U] 11.1.6 **weight**.

No weights are allowed if **twostep** is specified.

coeflegend does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Menu

Statistics > Treatment effects > Endogenous treatment estimators > Continuous outcome

Description

etregress estimates an average treatment effect (ATE) and the other parameters of a linear regression model augmented with an endogenous binary-treatment variable. Estimation is by either full maximum likelihood or a two-step consistent estimator.

In addition to the ATE, **etregress** can be used to estimate the average treatment effect on the treated (ATET) when the outcome may not be conditionally independent of the treatment.

etreg is a synonym for **etregress**.

Options for maximum likelihood estimates

Model

`treat(depart = indepvarst [, noconstant])` specifies the variables and options for the treatment equation. It is an integral part of specifying a treatment-effects model and is required.

`noconstant`, `constraints(constraints)`, `collinear`; see [R] [estimation options](#).

SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`oim`, `opg`), that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster clustvar`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce_option](#).

Reporting

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

`first` specifies that the first-step probit estimates of the treatment equation be displayed before estimation.

`noskip` specifies that a full maximum-likelihood model with only a constant for the regression equation be fit. This model is not displayed but is used as the base model to compute a likelihood-ratio test for the model test statistic displayed in the estimation header. By default, the overall model test statistic is an asymptotically equivalent Wald test that all the parameters in the regression equation are zero (except the constant). For many models, this option can substantially increase estimation time.

`hazard(newvar)` will create a new variable containing the hazard from the treatment equation. The hazard is computed from the estimated parameters of the treatment equation.

`nocnsreport`; see [R] [estimation options](#).

`display_options`: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `novlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `no!stretch`; see [R] [estimation options](#).

Maximization

`maximize_options`: `difficult`, `technique(algorithm_spec)`, `iterate(#)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, `nonrntolerance`, and `from(init_specs)`; see [R] [maximize](#). These options are seldom used.

Setting the optimization type to `technique(bhhh)` resets the default `vcetype` to `vce(opg)`.

The following option is available with `etregress` but is not shown in the dialog box:

`coeflegend`; see [R] [estimation options](#).

Options for two-step consistent estimates

Model

`treat(depart = indepvarst [, noconstant])` specifies the variables and options for the treatment equation. It is an integral part of specifying a treatment-effects model and is required.

`twostep` specifies that two-step consistent estimates of the parameters, standard errors, and covariance matrix be produced, instead of the default maximum likelihood estimates.

`noconstant`; see [R] [estimation options](#).

SE

`vce(vcetype)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`conventional`) and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce_option](#).

`vce(conventional)`, the default, uses the conventionally derived variance estimator for the two-step estimator of the treatment-effects model.

Reporting

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

`first` specifies that the first-step probit estimates of the treatment equation be displayed before estimation.

`hazard(newvar)` will create a new variable containing the hazard from the treatment equation. The hazard is computed from the estimated parameters of the treatment equation.

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

The following option is available with `etregress` but is not shown in the dialog box:

`coeflegend`; see [R] [estimation options](#).

Remarks and examples

Remarks are presented under the following headings:

[Overview](#)

[Basic example](#)

[Average treatment effect \(ATE\)](#)

[Average treatment effect on the treated \(ATET\)](#)

Overview

`etregress` estimates an ATE and the other parameters of a linear regression model that also includes an endogenous binary-treatment variable. In addition to the ATE, the parameters estimated by `etregress` can be used to estimate the ATET when the outcome is not conditionally independent of the treatment.

We call the model fit by `etregress` an endogenous treatment-regression model, although it is also known as an endogenous binary-variable model or as an endogenous dummy-variable model. The endogenous treatment-regression model is a specific endogenous treatment-effects model; it uses a linear model for the outcome and a constrained normal distribution to model the deviation from the conditional independence assumption imposed by the estimators implemented in `teffects`; see [TE] [teffects intro](#). In treatment-effects jargon, the endogenous binary-variable model is a linear potential-outcome model that allows for a specific correlation structure between the unobservables that affect the treatment and the unobservables that affect the potential outcomes. See [TE] [etpoisson](#) for an estimator that allows for a nonlinear outcome model and a similar model for the endogeneity of the treatment.

Heckman (1976, 1978) brought this model into the modern literature. Maddala (1983) derives the maximum likelihood and two-step estimators of the version implemented here, reviews some empirical applications of this model, and describes it as a constrained endogenous-switching model. Barnow, Cain, and Goldberger (1981) provide another useful derivation of this model. They concentrate on deriving the conditions for which the self-selection bias of the simple OLS estimator of the treatment effect, δ , is nonzero and of a specific sign. Cameron and Trivedi (2005, sec. 16.7 and 25.3.4) and Wooldridge (2010, sec. 21.4.1) discuss the endogenous binary-variable model as an endogenous treatment-effects model and link it to recent work.

More formally, the endogenous treatment-regression model is composed of an equation for the outcome y_j and an equation for the endogenous treatment t_j ,

$$y_j = \mathbf{x}_j\boldsymbol{\beta} + \delta t_j + \epsilon_j$$

$$t_j = \begin{cases} 1, & \text{if } \mathbf{w}_j\boldsymbol{\gamma} + u_j > 0 \\ 0, & \text{otherwise} \end{cases}$$

where \mathbf{x}_j are the covariates used to model the outcome, \mathbf{w}_j are the covariates used to model treatment assignment, and the error terms ϵ_j and u_j are bivariate normal with mean zero and covariance matrix

$$\begin{bmatrix} \sigma^2 & \rho\sigma \\ \rho\sigma & 1 \end{bmatrix}$$

The covariates \mathbf{x}_j and \mathbf{w}_j are unrelated to the error terms; in other words, they are exogenous.

Basic example

When there are no interactions between the treatment variable and the outcome covariates, `etregress` directly estimates the ATE and the ATET.

► Example 1

We estimate the ATE of being a union member on wages of women in 1972 from a nonrepresentative extract of the National Longitudinal Survey on young women who were ages 14–26 in 1968. We will use the variables `wage` (wage), `grade` (years of schooling completed), `smsa` (an indicator for living in an SMSA—standard metropolitan statistical area), `black` (an indicator for being African-American), `tenure` (tenure at current job), and `south` (an indicator for living in the South).

We use `etregress` to estimate the parameters of the endogenous treatment-regression model.

```
. use http://www.stata-press.com/data/r13/union3
(National Longitudinal Survey. Young Women 14-26 years of age in 1968)
. etregress wage age grade smsa black tenure, treat(union = south black tenure)
Iteration 0:  log likelihood = -3097.9871
Iteration 1:  log likelihood = -3052.5988
Iteration 2:  log likelihood = -3051.5789
Iteration 3:  log likelihood = -3051.575
Iteration 4:  log likelihood = -3051.575
Linear regression with endogenous treatment      Number of obs   =       1210
Estimator: maximum likelihood                  Wald chi2(6)    =       681.89
Log likelihood = -3051.575                     Prob > chi2     =       0.0000
```

	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
wage						
age	.1487409	.0193291	7.70	0.000	.1108566	.1866252
grade	.4205658	.0293577	14.33	0.000	.3630258	.4781058
smsa	.9117045	.1249041	7.30	0.000	.6668969	1.156512
black	-.7882471	.1367078	-5.77	0.000	-1.056189	-.5203047
tenure	.1524015	.0369596	4.12	0.000	.0799621	.2248409
union	2.945815	.2749624	10.71	0.000	2.406898	3.484731
_cons	-4.351572	.5283952	-8.24	0.000	-5.387208	-3.315936
union						
south	-.5807419	.0851111	-6.82	0.000	-.7475567	-.4139271
black	.4557499	.0958042	4.76	0.000	.2679772	.6435226
tenure	.0871536	.0232483	3.75	0.000	.0415878	.1327195
_cons	-.8855759	.0724506	-12.22	0.000	-1.027576	-.7435754
/athrho	-.6544344	.0910315	-7.19	0.000	-.8328529	-.4760159
/lnsigma	.7026768	.0293372	23.95	0.000	.645177	.7601767
rho	-.5746476	.0609711			-.6820049	-.4430472
sigma	2.01915	.0592362			1.906324	2.138654
lambda	-1.1603	.1495099			-1.453334	-.867266

```
LR test of indep. eqns. (rho = 0):  chi2(1) = 19.84  Prob > chi2 = 0.0000
```

All the covariates are statistically significant, and the likelihood-ratio test in the footer indicates that we can reject the null hypothesis of no correlation between the treatment errors and the outcome errors. The estimated ATE of being a union member is 2.95. The ATET is the same as the ATE in this case because the treatment indicator variable has not been interacted with any of the outcome covariates.

Although we discuss some details about this parameter below, the estimated correlation between the treatment-assignment errors and the outcome errors is -0.575 , indicating that unobservables that raise observed wages tend to occur with unobservables that lower union membership.

The results for the two ancillary parameters require explanation. For numerical stability during optimization, `etregress` does not directly estimate ρ or σ . Instead, `etregress` estimates the inverse hyperbolic tangent of ρ ,

$$\operatorname{atanh} \rho = \frac{1}{2} \ln \left(\frac{1 + \rho}{1 - \rho} \right)$$

and $\ln\sigma$. Also `etregress` reports $\lambda = \rho\sigma$, along with an estimate of the standard error of the estimate and the confidence interval.

◀

Average treatment effect (ATE)

When there is a treatment variable and outcome covariate interaction, the parameter estimates from `etregress` can be used by `margins` to estimate the ATE, the average difference of the treatment potential outcomes and the control potential outcomes.

▷ Example 2

In [example 1](#), the coefficients on the outcome covariates do not vary by treatment level. The differences in wages between union members and nonmembers are modeled as a level shift captured by the coefficient on the indicator for union membership. In this example, we use factor-variable notation to allow some of the coefficients to vary over treatment level and then use `margins` (see [\[R\] margins](#)) to estimate the ATE. (See [\[U\] 11.4.3 Factor variables](#) for an introduction to factor-variable notation.)

We begin by estimating the parameters of the model in which the coefficients on `black` and `tenure` differ for union members and nonmembers. We specify the `vce(robust)` option because we need to specify `vce(unconditional)` when we use `margins` below.

```

. etregress wage age grade smsa i.union#c.(black tenure),
> treat(union = south black tenure) vce(robust)
Iteration 0:  log pseudolikelihood = -3104.7035
Iteration 1:  log pseudolikelihood = -3053.2128
Iteration 2:  log pseudolikelihood = -3049.3217
Iteration 3:  log pseudolikelihood = -3049.2838
Iteration 4:  log pseudolikelihood = -3049.2838

Linear regression with endogenous treatment      Number of obs   =      1210
Estimator: maximum likelihood                  Wald chi2(8)    =      493.40
Log pseudolikelihood = -3049.2838             Prob > chi2     =      0.0000

```

	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
wage						
age	.1489075	.0207283	7.18	0.000	.1082809	.1895342
grade	.4200493	.0377621	11.12	0.000	.346037	.4940616
smsa	.9232614	.1201486	7.68	0.000	.6877746	1.158748
union#						
c.black						
0	-.6685584	.1444213	-4.63	0.000	-.9516189	-.3854979
1	-1.183101	.2574817	-4.59	0.000	-1.687756	-.6784459
union#						
c.tenure						
0	.1687459	.0503107	3.35	0.001	.0701387	.2673531
1	.0836366	.0903669	0.93	0.355	-.0934793	.2607526
union	3.342862	.5586856	5.98	0.000	2.247859	4.437866
_cons	-4.42566	.6493003	-6.82	0.000	-5.698265	-3.153054
union						
south	-.5844679	.0833069	-7.02	0.000	-.7477464	-.4211894
black	.4740688	.093241	5.08	0.000	.2913198	.6568179
tenure	.0874297	.0253892	3.44	0.001	.0376678	.1371916
_cons	-.8910483	.0746329	-11.94	0.000	-1.037326	-.7447705
/athrho						
/athrho	-.673316	.2215326	-3.04	0.002	-1.107512	-.23912
/lnsigma						
/lnsigma	.7055908	.0749711	9.41	0.000	.5586502	.8525315
rho						
rho	-.5871569	.1451585			-.8031811	-.2346644
sigma						
sigma	2.025043	.1518197			1.748311	2.345577
lambda						
lambda	-1.189018	.3631074			-1.900695	-.4773404

Wald test of indep. eqns. (rho = 0): chi2(1) = 22.35 Prob > chi2 = 0.0000

The results indicate that the coefficients on black differ by union membership and that the coefficient on tenure for nonmembers is positive, while the coefficient on tenure for members is 0. The model fits well overall, so we proceed with interpretation. Because we interacted the treatment variable with two of the covariates, the estimated coefficient on the treatment level is not an estimate of the ATE. Below we use margins to estimate the ATE from these results. We specify the `vce(unconditional)` option to obtain the standard errors for the population ATE instead of the sample ATE. We specify the `contrast(nowald)` option to suppress the Wald tests, which margins displays by default for contrasts.

```
. margins r.union, vce(unconditional) contrast(nowald)
Contrasts of predictive margins
Expression   : Linear prediction, predict()
```

	Unconditional			
	Contrast	Std. Err.	[95% Conf. Interval]	
union (1 vs 0)	3.042691	.5305143	2.002902	4.08248

The ATE estimate is essentially the same as the one produced by the constrained model in [example 1](#). ◀

Average treatment effect on the treated (ATET)

When there is a treatment variable and outcome covariate interaction, the parameter estimates from `etregress` can be used by `margins` to estimate the ATET, the average difference of the treatment potential outcomes and the control potential outcomes on the treated population.

▶ Example 3

The ATET may differ from the ATE in the [previous example](#) because the interaction between the treatment variable and some outcome covariates makes the ATE and the ATET vary over outcome covariate values. Below we use `margins` to estimate the ATET by specifying the `subpop(union)` option, which restricts the sample used by `margins` to union members.

```
. margins r.union, vce(unconditional) contrast(nowald) subpop(union)
Contrasts of predictive margins
Expression   : Linear prediction, predict()
```

	Unconditional			
	Contrast	Std. Err.	[95% Conf. Interval]	
union (1 vs 0)	2.96898	.5358449	1.918744	4.019217

The estimated ATET and ATE are close, indicating that the average predicted outcome for the treatment group is similar to the average predicted outcome for the whole population. ◀

Stored results

etregress (maximum likelihood) stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(k)</code>	number of parameters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_eq_model)</code>	number of equations in overall model test
<code>e(k_aux)</code>	number of auxiliary parameters
<code>e(k_dv)</code>	number of dependent variables
<code>e(df_m)</code>	model degrees of freedom
<code>e(ll)</code>	log likelihood
<code>e(ll_0)</code>	log likelihood, constant-only model (<code>noskip</code> only)
<code>e(N_clust)</code>	number of clusters
<code>e(lambda)</code>	λ
<code>e(selambda)</code>	standard error of λ
<code>e(sigma)</code>	estimate of sigma
<code>e(chi2)</code>	χ^2
<code>e(chi2_c)</code>	χ^2 for comparison test
<code>e(p_c)</code>	p -value for comparison test
<code>e(p)</code>	significance
<code>e(rho)</code>	ρ
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(rank0)</code>	rank of <code>e(V)</code> for constant-only model
<code>e(ic)</code>	number of iterations
<code>e(rc)</code>	return code
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	etregress
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(hazard)</code>	variable containing hazard
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(chi2type)</code>	Wald or LR; type of model χ^2 test
<code>e(chi2_ct)</code>	Wald or LR; type of model χ^2 test corresponding to <code>e(chi2_c)</code>
<code>e(vce)</code>	<i>vctype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. Err.
<code>e(opt)</code>	type of optimization
<code>e(which)</code>	<code>max</code> or <code>min</code> ; whether optimizer is to perform maximization or minimization
<code>e(method)</code>	<code>ml</code>
<code>e(ml_method)</code>	type of <code>ml</code> method
<code>e(user)</code>	name of likelihood-evaluator program
<code>e(technique)</code>	maximization technique
<code>e(properties)</code>	<code>b V</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(footnote)</code>	program used to implement the footnote display
<code>e(marginsok)</code>	predictions allowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(Cns)</code>	constraints matrix
<code>e(ilog)</code>	iteration log (up to 20 iterations)
<code>e(gradient)</code>	gradient vector
<code>e(V)</code>	variance-covariance matrix of the estimators
<code>e(V_modelbased)</code>	model-based variance

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

etregress (two-step) stores the following in $e()$:

Scalars

<code>e(N)</code>	number of observations
<code>e(df_m)</code>	model degrees of freedom
<code>e(lambda)</code>	λ
<code>e(selambda)</code>	standard error of λ
<code>e(sigma)</code>	estimate of sigma
<code>e(chi2)</code>	χ^2
<code>e(p)</code>	significance
<code>e(rho)</code>	ρ
<code>e(rank)</code>	rank of $e(V)$

Macros

<code>e(cmd)</code>	<code>etregress</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(title)</code>	title in estimation output
<code>e(chi2type)</code>	Wald or LR; type of model χ^2 test
<code>e(vce)</code>	<code>vce</code> type specified in <code>vce()</code>
<code>e(vctype)</code>	title used to label Std. Err.
<code>e(hazard)</code>	variable specified in <code>hazard()</code>
<code>e(method)</code>	<code>ml</code> or <code>twostep</code>
<code>e(properties)</code>	<code>b V</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(footnote)</code>	program used to implement the footnote display
<code>e(marginsok)</code>	predictions allowed by <code>margins</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

Methods and formulas

Maddala (1983, 117–122) derives both the maximum likelihood and the two-step estimator implemented here. Greene (2012, 890–894) also provides an introduction to the treatment-effects model. Cameron and Trivedi (2005, sections 16.7 and 25.3.4) and Wooldridge (2010, section 21.4.1) discuss the endogenous binary-variable model as an endogenous treatment-effects model and link it to recent work.

The primary regression equation of interest is

$$y_j = \mathbf{x}_j\beta + \delta t_j + \epsilon_j$$

where t_j is a binary-treatment variable that is assumed to stem from an unobservable latent variable:

$$t_j^* = \mathbf{w}_j\gamma + u_j$$

The decision to obtain the treatment is made according to the rule

$$t_j = \begin{cases} 1, & \text{if } t_j^* > 0 \\ 0, & \text{otherwise} \end{cases}$$

where ϵ and u are bivariate normal with mean zero and covariance matrix

$$\begin{bmatrix} \sigma^2 & \rho\sigma \\ \rho\sigma & 1 \end{bmatrix}$$

The likelihood function for this model is given in [Maddala \(1983, 122\)](#). [Greene \(2000, 180\)](#) discusses the standard method of reducing a bivariate normal to a function of a univariate normal and the correlation ρ . The following is the log likelihood for observation j ,

$$\ln L_j = \begin{cases} \ln \Phi \left\{ \frac{\mathbf{w}_j \boldsymbol{\gamma} + (y_j - \mathbf{x}_j \boldsymbol{\beta} - \delta) \rho / \sigma}{\sqrt{1 - \rho^2}} \right\} - \frac{1}{2} \left(\frac{y_j - \mathbf{x}_j \boldsymbol{\beta} - \delta}{\sigma} \right)^2 - \ln(\sqrt{2\pi}\sigma) & t_j = 1 \\ \ln \Phi \left\{ \frac{-\mathbf{w}_j \boldsymbol{\gamma} - (y_j - \mathbf{x}_j \boldsymbol{\beta}) \rho / \sigma}{\sqrt{1 - \rho^2}} \right\} - \frac{1}{2} \left(\frac{y_j - \mathbf{x}_j \boldsymbol{\beta}}{\sigma} \right)^2 - \ln(\sqrt{2\pi}\sigma) & t_j = 0 \end{cases}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.

In the maximum likelihood estimation, σ and ρ are not directly estimated. Rather $\ln \sigma$ and $\operatorname{atanh} \rho$ are directly estimated, where

$$\operatorname{atanh} \rho = \frac{1}{2} \ln \left(\frac{1 + \rho}{1 - \rho} \right)$$

The standard error of $\lambda = \rho\sigma$ is approximated through the delta method, which is given by

$$\operatorname{Var}(\lambda) \approx \mathbf{D} \operatorname{Var}\{(\operatorname{atanh} \rho \ \ln \sigma)\} \mathbf{D}'$$

where \mathbf{D} is the Jacobian of λ with respect to $\operatorname{atanh} \rho$ and $\ln \sigma$.

With maximum likelihood estimation, this command supports the Huber/White/sandwich estimator of the variance and its clustered version using `vce(robust)` and `vce(cluster clustvar)`, respectively. See [\[P\] `_robust`](#), particularly [Maximum likelihood estimators](#) and [Methods and formulas](#).

The maximum likelihood version of `etregress` also supports estimation with survey data. For details on VCEs with survey data, see [\[SVY\] variance estimation](#).

[Maddala \(1983, 120–122\)](#) also derives the two-step estimator. In the first stage, probit estimates are obtained of the treatment equation

$$\Pr(t_j = 1 \mid \mathbf{w}_j) = \Phi(\mathbf{w}_j \boldsymbol{\gamma})$$

From these estimates, the hazard, h_j , for each observation j is computed as

$$h_j = \begin{cases} \phi(\mathbf{w}_j \hat{\boldsymbol{\gamma}}) / \Phi(\mathbf{w}_j \hat{\boldsymbol{\gamma}}) & t_j = 1 \\ -\phi(\mathbf{w}_j \hat{\boldsymbol{\gamma}}) / \{1 - \Phi(\mathbf{w}_j \hat{\boldsymbol{\gamma}})\} & t_j = 0 \end{cases}$$

where ϕ is the standard normal density function. If

$$d_j = h_j(h_j + \hat{\boldsymbol{\gamma}} \mathbf{w}_j)$$

then

$$E(y_j | t_j, \mathbf{x}_j, \mathbf{w}_j) = \mathbf{x}_j\beta + \delta t_j + \rho\sigma h_j$$

$$\text{Var}(y_j | t_j, \mathbf{x}_j, \mathbf{w}_j) = \sigma^2 (1 - \rho^2 d_j)$$

The two-step parameter estimates of β and δ are obtained by augmenting the regression equation with the hazard h . Thus the regressors become $[\mathbf{x} \ \mathbf{t} \ h]$, and the additional parameter estimate β_h is obtained on the variable containing the hazard. A consistent estimate of the regression disturbance variance is obtained using the residuals from the augmented regression and the parameter estimate on the hazard

$$\hat{\sigma}^2 = \frac{\mathbf{e}'\mathbf{e} + \beta_h^2 \sum_{j=1}^N d_j}{N}$$

The two-step estimate of ρ is then

$$\hat{\rho} = \frac{\beta_h}{\hat{\sigma}}$$

To understand how the consistent estimates of the coefficient covariance matrix based on the augmented regression are derived, let $\mathbf{A} = [\mathbf{x} \ \mathbf{t} \ h]$ and \mathbf{D} be a square diagonal matrix of size N with $(1 - \hat{\rho}^2 d_j)$ on the diagonal elements. The conventional VCE is

$$\mathbf{V}_{\text{twostep}} = \hat{\sigma}^2 (\mathbf{A}'\mathbf{A})^{-1} (\mathbf{A}'\mathbf{D}\mathbf{A} + \mathbf{Q}) (\mathbf{A}'\mathbf{A})^{-1}$$

where

$$\mathbf{Q} = \hat{\rho}^2 (\mathbf{A}'\mathbf{D}\mathbf{A}) \mathbf{V}_p (\mathbf{A}'\mathbf{D}\mathbf{A})$$

and \mathbf{V}_p is the variance–covariance estimate from the probit estimation of the treatment equation.

References

- Barnow, B. S., G. G. Cain, and A. S. Goldberger. 1981. Issues in the analysis of selectivity bias. In Vol. 5 of *Evaluation Studies Review Annual*, ed. E. W. Stromsdorfer and G. Farkas, 123–126. Beverly Hills: Sage.
- Brave, S., and T. Walstrum. 2014. Estimating marginal treatment effects using parametric and semiparametric methods. *Stata Journal* 14: 191–217.
- Brown, G. K., and T. Mergoupis. 2011. Treatment interactions with nonexperimental data in Stata. *Stata Journal* 11: 545–555.
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cong, R., and D. M. Drukker. 2000. sg141: Treatment effects model. *Stata Technical Bulletin* 55: 25–33. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, pp. 159–169. College Station, TX: Stata Press.
- Greene, W. H. 2000. *Econometric Analysis*. 4th ed. Upper Saddle River, NJ: Prentice Hall.
- . 2012. *Econometric Analysis*. 7th ed. Upper Saddle River, NJ: Prentice Hall.
- Heckman, J. 1976. The common structure of statistical models of truncation, sample selection and limited dependent variables and a simple estimator for such models. *Annals of Economic and Social Measurement* 5: 475–492.
- . 1978. Dummy endogenous variables in a simultaneous equation system. *Econometrica* 46: 931–959.
- Maddala, G. S. 1983. *Limited-Dependent and Qualitative Variables in Econometrics*. Cambridge: Cambridge University Press.
- Nannicini, T. 2007. Simulation-based sensitivity analysis for matching estimators. *Stata Journal* 7: 334–350.
- Nichols, A. 2007. Causal inference with observational data. *Stata Journal* 7: 507–541.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

[TE] **etregress postestimation** — Postestimation tools for etregress

[TE] **etpoisson** — Poisson regression with endogenous treatment effects

[R] **heckman** — Heckman selection model

[R] **probit** — Probit regression

[R] **regress** — Linear regression

[SVY] **svy estimation** — Estimation commands for survey data

[U] **20 Estimation and postestimation commands**

Title

etregress postestimation — Postestimation tools for etregress

[Description](#) [Syntax for predict](#) [Menu for predict](#) [Options for predict](#)
[Remarks and examples](#) [Also see](#)

Description

The following postestimation commands are available after `etregress`:

Command	Description
<code>contrast</code>	contrasts and ANOVA-style joint tests of estimates
<code>estat ic</code> ¹	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estat (svy)</code>	postestimation statistics for survey data
<code>estimates</code>	cataloging estimation results
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of coefficients
<code>lrtest</code> ²	likelihood-ratio test
<code>margins</code>	marginal means, predictive margins, marginal effects, and average marginal effects
<code>marginsplot</code>	graph the results from margins (profile plots, interaction plots, etc.)
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
<code>predict</code>	predictions, residuals, influence statistics, and other diagnostic measures
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>pwcompare</code>	pairwise comparisons of estimates
<code>suest</code> ¹	seemingly unrelated estimation
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

¹ `estat ic` and `suest` are not appropriate after `etregress`, `twostep`.

² `lrtest` is not appropriate with `svy` estimation results.

Syntax for predict

After ML or twostep

```
predict [type] newvar [if] [in] [, statistic]
```

After ML

```
predict [type] { stub* | newvarreg newvartreat newvarathrho newvarlnsigma }  
[if] [in] , scores
```

<i>statistic</i>	Description
------------------	-------------

Main	
xb	linear prediction; the default
stdp	standard error of the prediction
stdf	standard error of the forecast
yctr	$E(y_j \mid \text{treatment} = 1)$
ycntr	$E(y_j \mid \text{treatment} = 0)$
ptr	$\Pr(\text{treatment} = 1)$
xbtr	linear prediction for treatment equation
stdptr	standard error of the linear prediction for treatment equation

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

`stdf` is not allowed with `svy` estimation results.

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

xb, the default, calculates the linear prediction, $\mathbf{x}_j\mathbf{b}$.

stdp calculates the standard error of the prediction, which can be thought of as the standard error of the predicted expected value or mean for the observation's covariate pattern. The standard error of the prediction is also referred to as the standard error of the fitted value.

stdf calculates the standard error of the forecast, which is the standard error of the point prediction for one observation. It is commonly referred to as the standard error of the future or forecast value. By construction, the standard errors produced by **stdf** are always larger than those produced by **stdp**; see *Methods and formulas* in [R] **regress postestimation**.

yctr calculates the expected value of the dependent variable conditional on the presence of the treatment: $E(y_j \mid \text{treatment} = 1)$.

ycntr calculates the expected value of the dependent variable conditional on the absence of the treatment: $E(y_j \mid \text{treatment} = 0)$.

ptr calculates the probability of the presence of the treatment:
 $\Pr(\text{treatment} = 1) = \Pr(\mathbf{w}_j\boldsymbol{\gamma} + u_j > 0)$.

`xbtrt` calculates the linear prediction for the treatment equation.

`stdptrt` calculates the standard error of the linear prediction for the treatment equation.

`scores`, not available with `twostep`, calculates equation-level score variables.

The first new variable will contain $\partial \ln L / \partial (x_j \beta)$.

The second new variable will contain $\partial \ln L / \partial (w_j \gamma)$.

The third new variable will contain $\partial \ln L / \partial \operatorname{atanh} \rho$.

The fourth new variable will contain $\partial \ln L / \partial \ln \sigma$.

Remarks and examples

The average treatment effect (ATE) and the average treatment effect on the treated (ATET) are the parameters most frequently estimated by postestimation techniques after `etregress`.

When there are no interactions between the treatment variable and the outcome covariates, `etregress` directly estimates the ATE and the ATET; see [example 1](#) of [\[TE\] etregress](#).

When there are interactions between the treatment variable and the outcome covariates, you can use `margins` after `etregress` to estimate the ATE or the ATET. See [example 2](#) of [\[TE\] etregress](#) for an example of ATE estimation. See [example 3](#) of [\[TE\] etregress](#) for an example of ATET estimation.

Also see

[\[TE\] etregress](#) — Linear regression with endogenous treatment effects

[\[U\] 20 Estimation and postestimation commands](#)

Title

teffects — Treatment-effects estimation for observational data

[Syntax](#) [Description](#) [Also see](#)

Syntax

```
teffects subcommand ... [ , options ]
```

<i>subcommand</i>	Description
aipw	augmented inverse-probability weighting
ipw	inverse-probability weighting
ipwra	inverse-probability-weighted regression adjustment
nnmatch	nearest-neighbor matching
overlap	overlap plots
psmatch	propensity-score matching
ra	regression adjustment

Description

`teffects` estimates potential-outcome means (POMs), average treatment effects (ATEs), and average treatment effects on the treated (ATETs) using observational data. Regression-adjustment, inverse-probability-weighting, and matching estimators are provided, as are doubly robust methods that combine regression adjustment and inverse-probability weighting. `teffects overlap` plots the estimated densities of the probability of getting each treatment level.

The outcome models can be continuous, binary, count, or nonnegative. The treatment model can be binary, or it can be multinomial, allowing for multivalued treatments.

For a brief description and example of each estimator, see [Remarks and examples](#) in [\[TE\] teffects intro](#).

Also see

[\[TE\] teffects intro](#) — Introduction to treatment effects for observational data

Title

teffects intro — Introduction to treatment effects for observational data

[Description](#)

[Remarks and examples](#)

[Reference](#)

[Also see](#)

Description

This entry provides a nontechnical introduction to treatment-effects estimators and the `teffects` command in Stata. Advanced users may want to instead read [\[TE\] teffects intro advanced](#) or skip to the individual commands' entries.

The `teffects` command estimates potential-outcome means (POMS), average treatment effects (ATEs), and average treatment effects among treated subjects (ATETs) using observational data.

Treatment effects can be estimated using regression adjustment (RA), inverse-probability weights (IPW), and “doubly robust” methods, including inverse-probability-weighted regression adjustment (IPWRA) and augmented inverse-probability weights (AIPW), and via matching on the propensity score or nearest neighbors.

The outcome models can be continuous, binary, count, or nonnegative. Continuous outcomes can be modeled using linear regression; binary outcomes can be modeled using logit, probit, or heteroskedastic probit regression; and count and nonnegative outcomes can be modeled using Poisson regression. The treatment model can be binary or multinomial. Binary treatments can be modeled using logit, probit, or heteroskedastic probit regression, while multinomial outcomes are modeled using multinomial logit regression.

Remarks and examples

This entry presents a nontechnical overview of treatment-effects estimators for those who are new to the subject of treatment-effects estimation or are at least new to Stata's facilities for estimating treatment effects. More advanced users may want to instead read [\[TE\] teffects intro advanced](#) or skip to the individual commands' entries.

Remarks are presented under the following headings:

[Introduction](#)

[Defining treatment effects](#)

[Estimating treatment effects](#)

[A quick tour of the estimators](#)

[Video example](#)

Introduction

Suppose we have observed a sample of subjects, some of whom received a treatment and the rest of whom did not. As the name suggests, in most applications, the “subjects” are indeed people. A “treatment” could indeed be a medical treatment such as a new drug regimen or surgical procedure. In social science applications, a treatment could be participation in a job-training program or inclusion in a classroom or school in which a new pedagogical method is being used. However, not all applications use individuals as the subjects. For example, a policy analyst might be interested in examining the impact of an experimental program in which a national agency held a lottery to award only some local governments the resources needed to implement the program. Here the subjects are the local governments, and treatment refers to whether a local government received the resources needed to implement the program.

We would like to know if a treatment has an effect on an outcome Y . The outcome could be the cholesterol level of a patient taking either an existing statin or a new experimental drug, or the outcome could be the wage offered to a person who either did or did not participate in a job-training program. In an ideal world, we would observe Y when a subject is treated (which we denote as Y_1), and we would observe Y when the same subject is not treated (which we denote as Y_0). We would be careful to make both observations under identical conditions so that the only difference is the presence or absence of the treatment. We could then average the difference between Y_1 and Y_0 across all the subjects in our dataset to obtain a measure of the average impact of the treatment.

Unfortunately, this ideal experiment is almost never available in observational data because it is not possible to observe a specific subject having received the treatment and having not received the treatment. When the outcome is the birthweight of a specific baby and the treatment is the mother smoking while pregnant, it is impossible to observe the baby's birthweight under both treatments of the mother smoking and the mother not smoking.

A classic solution to this problem is to randomize the treatment. High costs or ethical issues rule out this solution in many observational datasets. For example, we could not ask a random selection of pregnant women to smoke.

The defining characteristic of observational data is that treatment status is not randomized. Moreover, that implies that the outcome and treatment are not necessarily independent. The goal of the estimators implemented by `teffects` is to utilize covariates to make treatment and outcome independent once we condition on those covariates.

The treatment-effect estimators implemented by `teffects` allow us to estimate the efficacy of treatments using observational data. The rest of this entry discusses these treatment-effect estimators at an introductory level. For a more technical introduction, see [\[TE\] `teffects intro advanced`](#).

Defining treatment effects

We introduce treatment effects more formally by using the potential-outcomes framework, which is also known as the counterfactual framework. What is a potential outcome? Consider a subject that did not receive treatment so that we observe Y_0 . What would Y_1 be for that same subject if it were exposed to treatment? We call Y_1 the potential outcome or counterfactual for that subject. For a subject that did receive treatment, we observe Y_1 , so Y_0 would be the counterfactual outcome for that subject. We can view this as a missing-data problem, and treatment-effect methods can account for that problem.

Treatment-effect estimators allow us to estimate three parameters. The potential-outcome means (POMs) are the means of Y_1 and Y_0 in the population. The average treatment effect (ATE) is the mean of the difference ($Y_1 - Y_0$). Finally, the average treatment effect on the treated (ATET) is the mean of the difference ($Y_1 - Y_0$) among the subjects that actually receive the treatment.

To develop our intuition, suppose we have observed a sample of patients, some of whom received a medication to reduce their blood pressure. [Figure 1](#) plots each of our patient's systolic blood pressures as a function of weight. We use the color green to indicate patients who did not receive the drug and blue to indicate patients who did receive the drug.

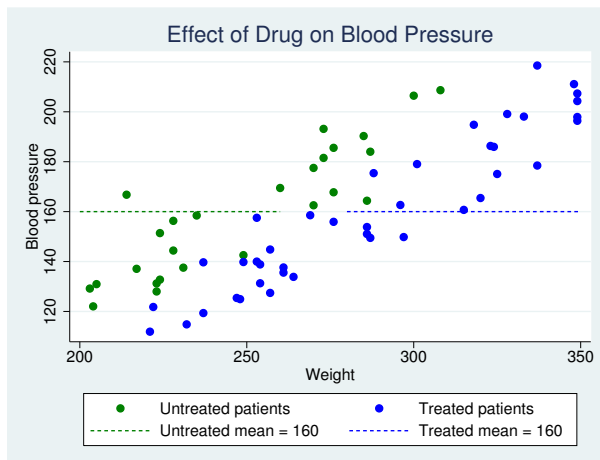


Figure 1

A remarkable feature of our data is that the average blood pressure of patients not taking the drug is 160, and the average blood pressure of patients taking the drug is also 160. Can we therefore conclude that taking the drug has no impact on blood pressure? The answer is no.

Because this is observational data, we could not randomly assign who would receive the drug and who would not. As a result, treatment status could be related to covariates that also affect blood pressure. Heavier patients were more likely to be prescribed the medication, and blood pressure is correlated with weight. The difference in sample means does not estimate the true average treatment effect, because blood pressure depends on weight and weight is correlated with the treatment.

Suppose that we did in fact observe both potential outcomes for all patients. In [figure 2](#), we continue to use solid dots for our observed data points, and we introduce hollow dots to represent the counterfactual outcomes. That is, the green hollow dots represent the blood pressures we would measure if only our treated patients had not taken the drug, and the blue hollow dots represent the blood pressures we would measure if only our untreated patients had taken the drug. The green and blue dashed lines represent the untreated and treated POMs, respectively. That is, the green line represents the mean of all the green dots, and the blue line represents the mean of all the blue dots.

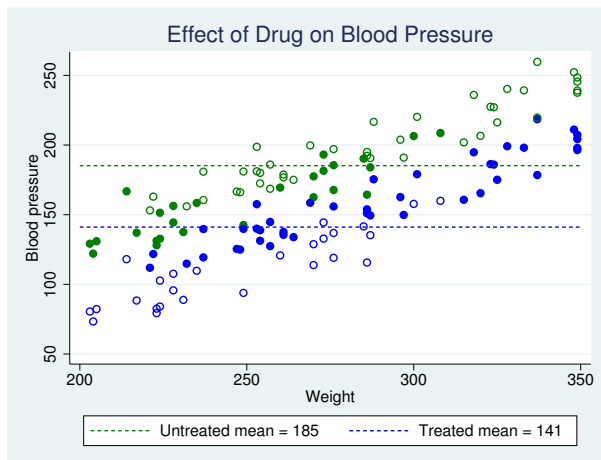


Figure 2

If we did have the data represented by the hollow dots, then we could say that the average treatment effect is the difference between the mean of all the green dots and the mean of all the blue dots. In this ideal scenario, there are no missing data on the other potential outcome, and we have all the data we need to use the difference in means to estimate the ATE.

Looking at figure 2, we can see why a difference in means using only the solid dots does not estimate the ATE. Using only the solid green dots underestimates the average blood pressure for untreated individuals, and using only the solid blue dots overestimates the average blood pressure for treated individuals.

Estimating an ATE is essentially a missing-data problem. When covariates that affect the potential outcomes are related to treatment, we cannot use a difference in sample means, because the missing data are informative.

The treatment-effect estimators implemented in `teffects` allow for covariates like weight to be related to the potential outcomes and the treatment. Essentially, the estimators implemented by `teffects` utilize covariates to fill in the hollow circles or otherwise account for how the missing data depend on covariates that affect the potential outcomes.

Estimating treatment effects

We cannot estimate the ATE by simply taking the difference between the sample means for the treated and untreated subjects, because there are covariates that are related to the potential outcomes and the treatment. The estimators implemented by `teffects` require us to specify enough of these covariates so that after we condition on these covariates, any remaining influences on the treatment are not related to the potential outcomes. `teffects` implements several different estimators to accomplish this, including regression adjustment (RA), inverse-probability weighting (IPW), “doubly robust” methods that combine elements of RA and IPW, and matching methods. Here we introduce the methods by using intuition and simple examples.

See [\[TE\] `teffects intro advanced`](#) for a more technical introduction, and see the individual commands’ entries for estimator-specific details.

Regression adjustment

The RA method extends the idea of using sample means to estimate treatment effects by using a regression model to predict potential outcomes adjusted for covariates. In the examples here, we use linear regression, but the `teffects ra` command provides you with the flexibility to use logistic, probit, and heteroskedastic probit regression models for binary outcomes as well as Poisson regression for nonnegative outcomes; see [TE] [teffects ra](#) for more information.

`bweightex.dta` is a hypothetical dataset based on [Cattaneo \(2010\)](#) that we have created to illustrate treatment-effects estimators using graphs. The subjects in this dataset are women who were pregnant, some of whom smoked during the pregnancy. The outcome variable is the birthweight of the baby, and we want to know whether smoking during pregnancy affects the birthweight. The dataset also contains other demographic variables that we will use later.

Figure 3 illustrates the relationship between birthweight and smoking status as a function of the mother's age:

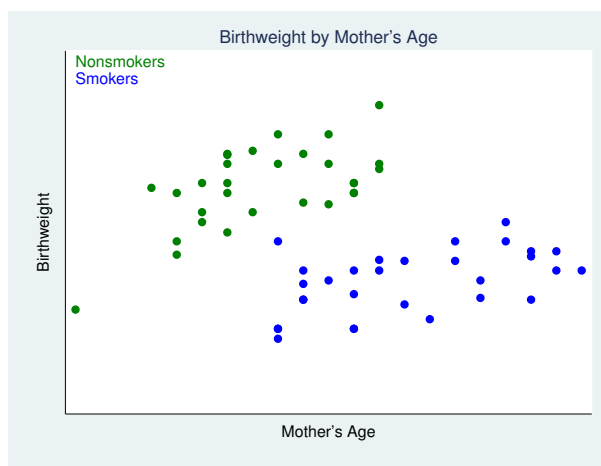


Figure 3

We see that smokers tend to be older than nonsmokers and that birthweight depends on smoking. Therefore, the difference between the sample means of birthweights of babies born to smokers and nonsmokers will not estimate the true average treatment effect.

We also still have the same problem as in the previous section: we do not observe the counterfactual birthweights of babies. Suppose, however, that we did. In [figure 4](#), we use solid points to represent observed birthweights and the colors green to represent nonsmokers and blue to represent smokers. The hollow points represent the counterfactual birthweights. The hollow blue points represent the birthweights of babies that we would observe if only our young nonsmoking mothers had instead smoked during their pregnancies. Similarly, the hollow green points represent the birthweights of babies that we would observe if only our older smoking mothers had instead not smoked during their pregnancies.

Figure 4 suggests a way to estimate the potential outcomes for each mother:

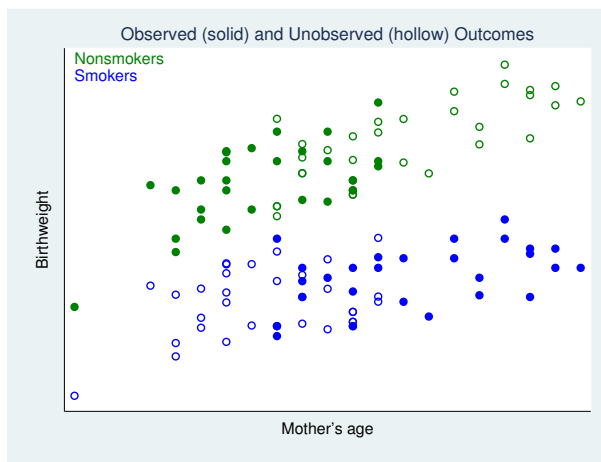


Figure 4

We could fit a linear regression of birthweight on mother's age by using the observed birthweights for nonsmokers, and we could do likewise for smokers. The following graph includes these two regression lines:

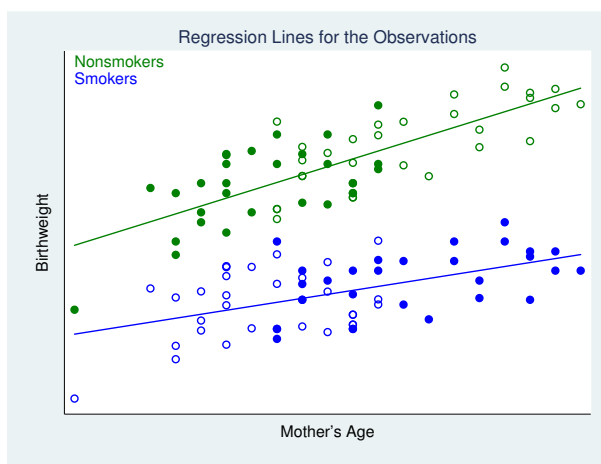


Figure 5

Figure 5 illustrates the principle behind the RA method. We use the green regression line to predict each baby's birthweight assuming the mother did not smoke, and we use the blue regression line to predict each baby's birthweight assuming the mother did smoke. The treatment effect of smoking for a mother of a particular age is the vertical difference between the green and blue regression lines.

The three parameters we mentioned in the introduction are now easy to estimate. For each mother, we obtain two values, say, bw_0 and bw_1 , representing our predictions of her baby's birthweight assuming the mother did not or did smoke, respectively. The means of these variables represent the untreated and treated POMs. The ATE is the sample mean of the difference $(bw_1 - bw_0)$, and the ATET is the sample mean of that difference computed using only the mothers who in fact did smoke during pregnancy.

Adding the circles highlights the fact that the average age is higher for smokers than for nonsmokers. Even though the blue and green lines have different slopes, if the average age was the same for smokers and nonsmokers, a difference in the sample means of birthweights could still estimate the true ATE.

Figure 5 lets us address one more issue. Users who are versed in regression analysis may be inclined to estimate the effect of smoking using a regression model for birthweight as a function of smoking and the mother's age. We clearly see in figure 5 that regression lines for smokers and nonsmokers have different slopes—the effect of age on birthweight is not the same for smokers and nonsmokers. In regression analysis, we would therefore include an interaction term between smoking and age. The RA method fits separate regression lines for smokers and nonsmokers, which also handles these differential effects of age on smoking.

Inverse-probability weighting

As we remarked in our discussion of the RA method, we cannot simply use the sample mean birthweights of babies born to smokers and nonsmokers to estimate the effect of smoking. If we did that, we would conflate the negative effect of smoking with the positive effect of age and the positive relationship between age and smoking. IPW is a treatment-effects estimator that uses weighted means rather than simple unweighted means to disentangle the effects of treatment and other confounders like age.

The concept underlying IPW can be gleaned from figure 2, where, as you will recall, the hollow points represent counterfactual outcomes. As we demonstrated in *Defining treatment effects*, we could estimate the average treatment effect if we knew the means of all the nonsmoking outcomes and the means of all the smoking outcomes. In the context of figure 4, we need the mean of all the green points, both solid and hollow, and the mean of all the blue points, both solid and hollow. Then the ATE would be the difference between those two means. IPW estimators view the hollow circles as missing data and correct the estimators for treated and not treated sample means for the missing data.

If we tried to calculate the mean nonsmoking birthweight based on just the solid green points, that mean would be biased downward because we are ignoring all the hollow green points, which correspond to higher birthweights. In IPW, we therefore apply more weight to the solid green points corresponding to older mothers and less weight to those corresponding to younger mothers. Using this weighting scheme will pull up the estimated mean birthweight of babies born to nonsmoking mothers to better approximate the true mean of all nonsmoking outcomes. The method for the mean smoking birthweight is virtually the same: we need to apply more weight to the younger smoking mothers than to the older smoking mothers to better approximate the true mean of all smoking outcomes.

Where do these weights for the weighted means come from? As the name implies, IPW uses the reciprocal (inverse) of the probability of being in the observed treatment group. These probabilities are obtained by fitting a model of treatment status on whatever characteristics we have on each subject. In our exposition of the RA method, we focused solely on the mother's age and smoking status as determinants of each baby's birthweight, mainly so that we could provide a graphical analysis. In the same vein, we could fit a logistic regression model of smoking status using only age to predict the probability that each mother smoked during pregnancy. However, the utility of the IPW method depends on how accurately the treatment model predicts the probability of treatment. In our dataset, we have other variables such as alcohol consumption, marital status, and the education level of the baby's father that are likely to help explain whether the mother smoked during pregnancy. As an added bonus, we can still provide a graphical exposition of IPW even with these additional explanatory variables.

We first fit a logistic model of the mother's smoking status, `mbsmoke`, as a function of marital status (`mmarried`), alcohol consumption (`alcohol`), the education level of the father (`fedu`), and the mother's age (`mage`):

```
. use http://www.stata-press.com/data/r13/bweightex
(Hypothetical birthweight data)
. logistic mbsmoke mmarried alcohol mage fedu
Logistic regression
```

	Number of obs	=	60
	LR chi2(4)	=	46.50
	Prob > chi2	=	0.0000
	Pseudo R2	=	0.5590

```
Log likelihood = -18.339432
```

mbsmoke	Odds Ratio	Std. Err.	z	P> z	[95% Conf. Interval]	
<code>mmarried</code>	.0785086	.0909212	-2.20	0.028	.0081122	.7597976
<code>alcohol</code>	18.81727	27.98003	1.97	0.048	1.020649	346.9259
<code>mage</code>	2.147569	.459327	3.57	0.000	1.41218	3.265909
<code>fedu</code>	.8189843	.1157528	-1.41	0.158	.6208252	1.080393
<code>_cons</code>	4.46e-07	2.12e-06	-3.07	0.002	3.96e-11	.0050329

Next we compute the inverse-probability weights, which we will store in a variable called `ps`. In the IPW method, for subjects who did receive treatment, the weight is equal to the reciprocal of the predicted probability of treatment. For subjects who did not receive treatment, the weight is equal to the reciprocal of the predicted probability of not receiving treatment; the probability of not receiving treatment is just one minus the probability of receiving treatment:

```
. predict ps
(option pr assumed; Pr(mbsmoke))
. replace ps = 1/ps if mbsmoke==1
(30 real changes made)
. replace ps = 1/(1-ps) if mbsmoke==0
(30 real changes made)
```

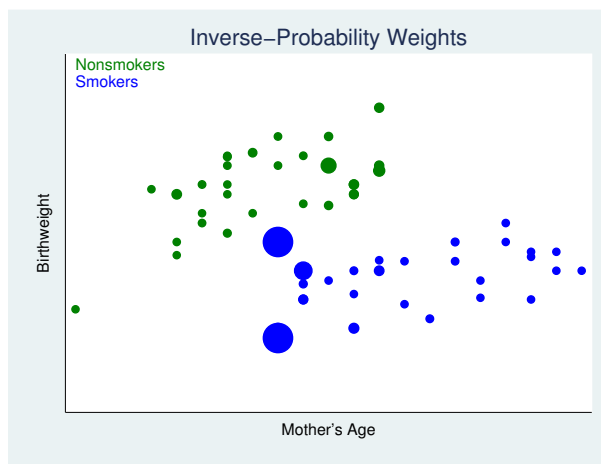


Figure 6

Figure 6 replicates [figure 3](#) with one twist. Rather than making all the points the same size, we have made the size of the points proportional to the IPW variable `ps`. Notice that the largest blue points correspond to the youngest smoking mothers in our sample, so they will receive the most weight when we compute the weighted mean birthweight of babies born to smoking mothers, just

as we explained we wanted to do. Similarly, the green points corresponding to older nonsmoking mothers are larger, representing larger weights.

There is a caveat to the IPW estimator. We remarked that in our sample, we had very few young mothers who smoked. When we fit our logistic or probit model and obtain the predicted probabilities, it is possible that some of the predictions are close to zero. The IPW is the reciprocal of that probability, and as the probability goes to zero, the weight becomes arbitrarily large. In those cases, the IPW can become unstable. This phenomenon is related to the concept of overlap, which means that every subject must have a positive probability of obtaining treatment. See [TE] [teffects overlap](#) and [TE] [teffects intro advanced](#) for more information about overlap.

Another limitation of the IPW estimator is that because we are using weighted means to estimate the POMs and ATE, unlike the RA estimator, we cannot obtain subject-level predictions of the treatment effects or potential outcomes. We do not have the two regression lines that we can use to predict outcomes for each subject.

Doubly robust combinations of RA and IPW

You may have noticed a clear distinction between the RA and IPW estimators. In the case of RA, we built linear regression models to predict the outcomes (birthweights) of each subject but said nothing about how treatment (smoking) arises. In the case of IPW, we built a logistic regression model to predict treatment status but did not build a formal model of the outcome. Doubly robust estimators combine the outcome modeling strategy of RA and the treatment modeling strategy of IPW. These estimators have a remarkable property: although they require us to build two models, we only need to specify one of the two models correctly. If we misspecify the treatment model but correctly specify the outcome model, we still obtain correct estimates of the treatment effect. If we correctly specify the treatment model but misspecify the outcome model, we again will obtain correct estimates of the treatment effect.

Stata's `teffects` command implements two doubly robust estimators, the augmented inverse-probability-weighted (AIPW) estimator and the inverse-probability-weighted regression-adjustment (IPWRA) estimator. These estimators combine elements of RA and IPW to be more robust to misspecification.

The AIPW estimator is an IPW estimator that includes an augmentation term that corrects the estimator when the treatment model is misspecified. When the treatment model is correctly specified, the augmentation term vanishes as the sample size becomes large. Like the IPW, the AIPW does not perform well when the predicted treatment probabilities are too close to zero or one.

The IPWRA estimator is an RA estimator that uses estimated inverse-probability weights to correct the estimator when the regression function is misspecified. When the regression function is correctly specified, the weights do not affect the consistency of the estimator.

Matching

Matching estimators are based on the idea of comparing the outcomes of subjects that are as similar as possible with the sole exception of their treatment status. In our birthweight and smoking example, we could select a mother who smokes and select a mother of the same age who does not smoke and compare the birthweights of their infants. The data of each mother serve as the potential outcome for the other mother.

For a single covariate such as age, identifying a pair of comparable mothers is not difficult. If we have a second covariate that is categorical, such as race, we might still be able to identify pairs of mothers who are the same age and of the same race assuming our dataset is large enough. However,

once we consider covariates that are measured on continuous scales or allow for more than a few discrete ones, then finding identical matches is a challenge. The solution is to use what is called a similarity measure, which is a statistic that measures how “close” two observations are. **teffects** offers two methods to find comparable observations based on similarity measures: nearest-neighbor matching and propensity-score matching.

Nearest-neighbor matching (NNM) is accomplished by calculating the “distance” between pairs of observations with regard to a set of covariates and then “matching” each subject to comparable observations that are closest to it. For example, suppose we have a variable that records each subject’s annual income to the penny. Say one subject who received treatment had an income of \$69,234.21. The likelihood that our dataset has an untreated subject who also earned \$69,234.21 is nil. However, we can determine the difference between each untreated subject’s income and our treated subject’s income, then match our treated subject with the untreated subjects whose income differences are smallest. Measuring the distance between subjects when we have multiple covariates is no challenge. By default, **teffects** uses what is known as the Mahalanobis distance, which is really nothing more than the Pythagorean theorem adapted to handle the fact that covariates may be correlated and measured on different scales.

NNM does not use a formal model for either the outcome or the treatment status, but this flexibility comes at a price. When matching on more than one continuous covariate, the NNM estimator must be augmented with a bias-correction term. **teffects nmatch** uses a linear function of the covariates specified in the `biasadj()` option to remove the large-sample bias.

Propensity-score matching (PSM) is an alternative to NNM. PSM matches on the estimated predicted probabilities of treatment, known as the propensity scores. PSM does not require bias correction, because it uses a model for the treatment. If the treatment model is reasonably well specified, PSM will perform at least as well as NNM; see [\[TE\] teffects intro advanced](#).

Caveats and assumptions

To use the estimators implemented in **teffects**, we must make several assumptions about the process that generated our data. Different estimators and statistics may require slightly more or slightly less restrictive assumptions and may exhibit varying degrees of robustness to departures from these assumptions, but in general, all the estimators require some form of the following three assumptions.

The independent and identically distributed (i.i.d.) sampling assumption ensures that the outcome and treatment status of each individual are unrelated to the outcome and treatment status of all the other individuals in the population. Correlated data arising from hierarchical or longitudinal study designs do not meet this assumption.

The conditional-independence (CI) assumption means once we control for all observable variables, the potential outcomes are independent of treatment assignment. The easiest way to understand the CI assumption is to understand when it is violated. In our birthweight example, suppose mothers who did not smoke were more health conscious and consumed better prenatal diets than those who did smoke. Unless we explicitly controlled for health awareness or diet, our model would violate the CI assumption: the mother’s decision to smoke or not smoke would not be independent of the baby’s birthweight. If we did not control for health awareness, we would overstate the negative impact of smoking on birthweight. Babies born to mothers who smoke weigh less than babies born to nonsmoking mothers not just because of the effects of cigarettes but also because of poorer prenatal diets.

In a study examining the effect of a job-training program, the CI assumption requires that there not be any unobserved factors such as ambition or work ethic that influence both whether a person enrolls in the program and the wage received upon completion. To use the methods implemented by

the `teffects` estimators, we must have variables in our dataset that allow us to control for those types of factors.

We mentioned the third assumption, overlap, in our discussions of IPW. More formally, the overlap assumption states that each individual have a positive probability of receiving treatment. In our birthweight example, we noted that there were no observations on young smokers and older nonsmokers. Perhaps we just have an unlucky sample, but to accurately assess the impact of treatment using these methods, we must have overlap to accurately estimate the counterfactual birthweights. In the context of matching estimators, overlap essentially means that we can actually match treated subjects with similar nontreated subjects.

A quick tour of the estimators

The `teffects` command implements six estimators of treatment effects. We introduce each one by showing the basic syntax one would use to apply them to our birthweight example. See each command's entry for more information.

Regression adjustment

`teffects ra` implements the RA estimator. We 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`). We use linear regression (the default) to model `bweight`:

```
. use http://www.stata-press.com/data/r13/cattaneo2
. teffects ra (bweight mmarried mage prenatal1 fbaby) (mbsmoke)
```

Inverse-probability weighting

`teffects ipw` implements the IPW estimator. Here we estimate the effect of smoking by using a probit model to predict the mother's smoking behavior as a function of marital status, the mother's age, and indicators for first-trimester doctor's visits and firstborn status:

```
. teffects ipw (bweight) (mbsmoke mmarried mage prenatal1 fbaby, probit)
```

Inverse-probability-weighted regression adjustment

`teffects ipwra` implements the IPWRA estimator. We model the outcome, birthweight, as a linear function of marital status, the mother's age, and indicators for first-trimester doctor's visits and firstborn status. We use a logistic model (the default) to predict the mother's smoking behavior, using the same covariates as explanatory variables:

```
. teffects ipwra (bweight mmarried mage prenatal1 fbaby) ///
(mbsmoke mmarried mage prenatal1 fbaby)
```

Augmented inverse-probability weighting

`teffects aipw` implements the AIPW estimator. Here we use the same outcome- and treatment-model specifications as we did with the IPWRA estimator:

```
. teffects aipw (bweight mmarried mage prenatal1 fbaby) ///  
              (mbsmoke mmarried mage prenatal1 fbaby)
```

Nearest-neighbor matching

`teffects nnmatch` implements the NNM estimator. In this example, we match treated and untreated subjects based on marital status, the mother's age, the father's age, and indicators for first-trimester doctor's visits and firstborn status. We use the Mahalanobis distance based on the mother's and father's ages to find matches. We use exact matching on the other three variables to enforce the requirement that treated subjects are matched with untreated subjects who have the same marital status and indicators for first-trimester doctor's visits and firstborn statuses. Because we are matching on two continuous covariates, we request that `teffects nnmatch` include a bias-correction term based on those two covariates:

```
. teffects nnmatch (bweight mage fage) (mbsmoke),          ///  
                  ematch(prenatal1 mmarried fbaby) biasadj(mage fage)
```

Propensity-score matching

`teffects psmatch` implements the PSM estimator. Here we model the propensity score using a probit model, incorporating marital status, the mother's age, and indicators for first-trimester doctor's visits and firstborn status as covariates:

```
. teffects psmatch (bweight) (mbsmoke mmarried mage prenatal1 fbaby, probit)
```

Video example

[Introduction to treatment effects in Stata, part 1](#)

Reference

Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154.

Also see

[TE] [teffects](#) — Treatment-effects estimation for observational data

[TE] [teffects intro advanced](#) — Advanced introduction to treatment effects for observational data

Title

teffects intro advanced — Advanced introduction to treatment effects for observational data

Description
Also see

Remarks and examples

Acknowledgments

References

Description

This entry provides a technical overview of treatment-effects estimators and their implementation in Stata. Those who are new to treatment-effects estimation may want to instead see [\[TE\] teffects intro](#).

The `teffects` command estimates potential-outcome means (POMs), average treatment effects (ATES), and average treatment effects among treated subjects (ATETs) using observational data.

Treatment effects can be estimated using regression adjustment (RA), inverse-probability weights (IPW), and “doubly robust” methods, including inverse-probability-weighted regression adjustment (IPWRA) and augmented inverse-probability weights (AIPW), and via matching on the propensity score or nearest neighbors.

The outcome models can be continuous, binary, count, or nonnegative. Continuous outcomes can be modeled using linear regression; binary outcomes can be modeled using logit, probit, or heteroskedastic probit regression; and count and nonnegative outcomes can be modeled using Poisson regression. The treatment model can be binary or multinomial. Binary treatments can be modeled using logit, probit, or heteroskedastic probit regression, while multinomial outcomes are modeled using multinomial logit regression.

Remarks and examples

This entry presents a technical overview of treatment-effects estimators and their implementation in Stata. Users who are new to treatment-effects estimators for observational data should instead read [\[TE\] teffects intro](#).

Remarks are presented under the following headings:

- Introduction*
- Defining treatment effects*
- The potential-outcome model*
- Assumptions needed for estimation*
- Comparing the ATE and ATET*
- Overview of treatment-effect estimators*
- RA estimators*
- IPW estimators*
- AIPW estimators*
- IPWRA estimators*
- Nearest-neighbor matching estimators*
- Propensity-score matching estimators*
- Choosing among estimators*
- Model choice*

Introduction

The `teffects` commands estimate treatment effects from observed data. A treatment effect is the change in an outcome caused by a subject, often an individual, getting one treatment instead of another. We cannot estimate individual-level treatment effects, because we only observe each individual getting one or another treatment.

Potential-outcome models provide a solution to this missing-data problem and allow us to estimate the distribution of individual-level treatment effects. A potential-outcome model specifies the potential outcomes that each individual would obtain under each treatment level, the treatment assignment process, and the dependence of the potential outcomes on the treatment assignment process.

When the potential outcomes do not depend on the treatment levels, after conditioning on covariates, regression estimators, inverse-probability-weighted estimators, and matching estimators are commonly used.

What we call the potential-outcome model is also known as the Rubin causal model and the counterfactual model. See [Rubin \(1974\)](#); [Holland \(1986\)](#); [Robins \(1986\)](#); [Heckman \(1997\)](#); [Heckman and Navarro-Lozano \(2004\)](#); [Imbens \(2004\)](#); [Cameron and Trivedi \(2005, chap. 2.7\)](#); [Imbens and Wooldridge \(2009\)](#); and [Wooldridge \(2010, chap. 21\)](#) for more detailed discussions.

Defining treatment effects

Three parameters are often used to measure treatment effects: the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs). In this section, we define each of these terms and introduce the notation and parameters used in the rest of our discussion.

In the binary-treatment case, the two potential outcomes for each individual are y_{0i} and y_{1i} ; y_{0i} is the outcome that would be obtained if i does not get the treatment, and y_{1i} is the outcome that would be obtained if i gets the treatment. y_{0i} and y_{1i} are realizations of the random variables y_0 and y_1 . Throughout this entry, i subscripts denote realizations of the corresponding unsubscripted random variables. We do not discuss multivalued treatments here, because doing so only increases the number of parameters and notation required and detracts from the essential points; see [\[TE\] **teffects multivalued**](#) for information about multivalued treatments.

The parameters of interest summarize the distribution of the unobservable individual-level treatment effect $y_1 - y_0$. In defining the parameters, t denotes a random treatment, t_i denotes the treatment received by individual i , $t = 1$ is the treatment level, and $t = 0$ is the control level. Given this notation, we can now define our parameters of interest.

ATE The ATE is the average effect of the treatment in the population:

$$\text{ATE} = E(y_1 - y_0)$$

POM The POM for treatment level t is the average potential outcome for that treatment level:

$$\text{POM}_t = E(y_t)$$

ATET The ATET is the average treatment effect among those that receive the treatment:

$$\text{ATET} = E(y_1 - y_0 | t = 1)$$

For an illustration of these concepts, see [Defining treatment effects](#) in [\[TE\] **teffects intro**](#).

The potential-outcome model

Next we specify a potential-outcome model that serves as a touchstone for the rest of our discussion. The model described here generates data in which y_i is the observed outcome variable, t_i is the treatment variable, \mathbf{x}_i is a vector of covariates that affect the outcome, and \mathbf{w}_i is a vector of covariates that affect the treatment assignment. \mathbf{x}_i and \mathbf{w}_i may have elements in common.

This potential-outcome model specifies that the observed outcome variable y is y_0 when $t = 0$ and that y is y_1 when $t = 1$. Algebraically, we say that

$$y = (1 - t)y_0 + ty_1$$

The functional forms for y_0 and y_1 are

$$y_0 = \mathbf{x}'\beta_0 + \epsilon_0 \tag{1}$$

$$y_1 = \mathbf{x}'\beta_1 + \epsilon_1 \tag{2}$$

where β_0 and β_1 are coefficients to be estimated, and ϵ_0 and ϵ_1 are error terms that are not related to \mathbf{x} or \mathbf{w} . This potential-outcome model separates each potential outcome into a predictable component, $\mathbf{x}\beta_t$, and an unobservable error term, ϵ_t .

The treatment assignment process is

$$t = \begin{cases} 1 & \text{if } \mathbf{w}'\gamma + \eta > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where γ is a coefficient vector, and η is an unobservable error term that is not related to either \mathbf{x} or \mathbf{w} . The treatment assignment process is also separated into a predictable component, $\mathbf{w}'\gamma$, and an unobservable error term, η .

We emphasize six points about this model:

1. The observed data from this model contain y_i , t_i , \mathbf{w}_i , and \mathbf{x}_i . The data do not reveal both y_{0i} and y_{1i} for any given i .
2. The model for t determines how the data on y_0 and y_1 are missing.
3. The model separates the potential outcomes and treatment assignment into observable and unobservable components.
4. Whether η is independent of the vector (ϵ_0, ϵ_1) is essential in specifying the set of available estimators.
5. The coefficient vectors β_0 , β_1 , and γ are auxiliary parameters. We use estimates of these coefficient vectors to estimate the ATE, the POMs, and the ATET.
6. For notational simplicity, we represented y_0 and y_1 as linear functions. In practice, we can use other functional forms.

In specifying this potential-outcome model, we explicitly showed the functional forms for the potential outcomes and the treatment assignment process. To ease subsequent discussions, we refer to the set of functional forms for the potential outcomes as the “outcome model”, and we refer to the treatment assignment process as the “treatment model”.

Assumptions needed for estimation

As with any type of estimator, we must make some assumptions to use treatment-effects estimators. The particular assumptions we need for each estimator implemented by `teffects` and for each effect parameter vary, but some version of each of the following is required.

CI The conditional-independence CI assumption restricts the dependence between the treatment model and the potential outcomes.

Overlap The overlap assumption ensures that each individual could receive any treatment level.

i.i.d. The independent and identically distributed (i.i.d.) sampling assumption ensures that the potential outcomes and the treatment status of each individual are unrelated to the potential outcomes and treatment statuses of all other individuals in the population.

We now discuss each assumption in detail.

The CI assumption

After conditioning on covariates, when no unobservable variable affects both treatment assignment and the potential outcomes, the potential outcomes are conditionally independent of the treatment. In epidemiological jargon, there are no unmeasured confounders. In econometric jargon, we have selection on observables. If we observe enough covariates, the potential outcomes may indeed be conditionally independent of the treatment.

Intuitively, the CI assumption says that only the covariates \mathbf{x} affect both the treatment and the potential outcomes. Any other factors that affect the treatment must be independent of the potential outcomes, and any other factors that affect the potential outcomes must be independent of the treatment. Formally, the CI assumption states that, conditional on covariates \mathbf{x} , the treatment t is independent of the vector of potential outcomes $(y_0, y_1)'$.

The CI assumption allows us to estimate the effects by regression adjustment (RA) methods, inverse-probability-weighting (IPW) methods, methods that combine RA and IPW concepts, and matching methods. The data only reveal information about $E(y_0|\mathbf{x}, \mathbf{w}, t = 0)$ and $E(y_1|\mathbf{x}, \mathbf{w}, t = 1)$, but we are interested in an average of $E(y_0|\mathbf{x}, \mathbf{w})$ and $E(y_1|\mathbf{x}, \mathbf{w})$, where \mathbf{x} represents the outcome covariates and \mathbf{w} the treatment-assignment covariates. The CI assumption allows us to estimate $E(y_0|\mathbf{x}, \mathbf{w})$ and $E(y_1|\mathbf{x}, \mathbf{w})$ directly from the observations for which $E(y_0|\mathbf{x}, \mathbf{w}, t = 0)$ and $E(y_1|\mathbf{x}, \mathbf{w}, t = 1)$, respectively.

For our potential-outcome model presented in (1) through (3), the CI assumption can be viewed as a set of restrictions on the covariance matrix of the error terms. Suppose that the vector of unobservables $(\epsilon_0, \epsilon_1, \eta)$ is normally distributed

$$\begin{pmatrix} \epsilon_0 \\ \epsilon_1 \\ \eta \end{pmatrix} \sim N \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_0^2 & \rho_{01}\sigma_0\sigma_1 & \rho_{\eta 0}\sigma_0 \\ \rho_{01}\sigma_0\sigma_1 & \sigma_1^2 & \rho_{\eta 1}\sigma_1 \\ \rho_{\eta 0}\sigma_0 & \rho_{\eta 1}\sigma_1 & 1 \end{pmatrix} \right\} \quad (4)$$

where σ_0 is the standard deviation of ϵ_0 , ρ_{01} is the correlation between ϵ_0 and ϵ_1 , σ_1 is the standard deviation of ϵ_1 , $\rho_{\eta 0}$ is the correlation between ϵ_η and ϵ_0 , and $\rho_{\eta 1}$ is the correlation between ϵ_η and ϵ_1 . As is standard in the normally distributed latent-variable specification of a binary-dependent variable, we normalize the variance of ϵ_η to 1.

CI specifies that $\rho_{\eta 0} = \rho_{\eta 1} = 0$ so that we can write (4) as

$$\begin{pmatrix} \epsilon_0 \\ \epsilon_1 \\ \eta \end{pmatrix} \sim N \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_0^2 & \rho_{01}\sigma_0\sigma_1 & 0 \\ \rho_{01}\sigma_0\sigma_1 & \sigma_1^2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right\}$$

Writing the covariance matrix this way makes clear what we mean by conditional independence: unobserved shocks that affect whether a subject is treated do not have any effect on the potential outcomes, and unobserved shocks that affect a potential outcome do not affect treatment.

The command `teffects` implements estimators that require the CI assumption. See [TE] `etregress` and [TE] `etpoisson` for commands that handle two cases in which the CI assumption is replaced by precise specifications of the joint dependence among the unobservables. Brown and Mergoupis (2011) discuss the `itreatreg` command that extends [TE] `etregress`.

The CI assumption is also known as unconfoundedness and selection-on-observables in the literature. See Rosenbaum and Rubin (1983); Heckman (1997); Heckman and Navarro-Lozano (2004); Cameron and Trivedi (2005, sec. 25.2.1); Tsiatis (2006, sec. 13.3); Angrist and Pischke (2009, chap. 3); Imbens and Wooldridge (2009); and Wooldridge (2010, sec. 21.3). Some discussions with Stata commands can be found in Becker and Caliendo (2007), Nichols (2007), and Daniel, De Stavola, and Cousens (2011).

□ Technical note

In fact, full CI is stronger than what we need to estimate the ATE, the ATET, or the POMS. For the estimators implemented in `teffects`, we only need a conditional mean independence (CMI) assumption. Intuitively, the CMI assumption says that after accounting for the covariates \mathbf{x}_i , the treatment does not affect the conditional mean of each potential outcome. Formally, the CMI requires that $E(y_0|\mathbf{x}, t) = E(y_0|\mathbf{x})$ and that $E(y_1|\mathbf{x}, t) = E(y_1|\mathbf{x})$. The CMI assumption allows the conditional variance to depend on the treatment, while the CI assumption does not.

The CI assumption implies the CMI assumption, but not vice versa.

See Wooldridge (2010, sec. 21.2 and 21.3) for an excellent introduction to this topic, and see Cattaneo, Drukker, and Holland (2013) for some discussion of the multiple treatment case. □

The overlap assumption

The overlap assumption requires that each individual have a positive probability of receiving each treatment level. Formally, the overlap assumption requires that for each possible \mathbf{x} in the population and each treatment level \tilde{t} , $0 < \Pr(t = \tilde{t}|\mathbf{x}) < 1$. Rosenbaum and Rubin (1983) call the combination of the CI and overlap assumptions strong ignorability; see also Abadie and Imbens (2006, 237–238) and Imbens and Wooldridge (2009).

The i.i.d. assumption

The third of the three assumptions listed above is the i.i.d. assumption; it is the standard assumption of having an i.i.d. sample from the population. In potential-outcome models, i.i.d. sampling implies that the potential outcomes and treatment status of each individual are unrelated to the potential outcomes and treatment statuses of all the other individuals in the population. I.i.d. sampling rules out interactions among the individuals. For instance, models of vaccinations in epidemiology and general

equilibrium effects in economics violate the independence assumption. This third assumption is a part of what is known as the stable unit treatment value assumption (SUTVA); see [Wooldridge \(2010, 905\)](#) and [Imbens and Wooldridge \(2009\)](#).

Comparing the ATE and ATET

When comparing the ATE and the ATET, two points should be mentioned.

First, the assumptions required to estimate the ATET are less restrictive than the assumptions required to estimate the ATE. Estimating the ATET requires a weaker form of the CI assumption and a weaker version of the overlap assumption.

To estimate the ATE under CI, we require that the unobservables in the treatment model be conditionally independent of the unobservables in both potential outcomes. In contrast, we can estimate the ATET under CI when the unobservables in the treatment model are conditionally independent of just the control-level potential outcome; see [Wooldridge \(2010, 906–912\)](#).

Although the ATE version of overlap requires that all covariate patterns have a positive probability of being allocated to each treatment state, we can estimate the ATET when only the covariate patterns for which someone is treated have a positive probability of being allocated to each treatment state. This weaker overlap assumption can be important in some studies. For example, [Heckman \(1997\)](#) discusses how the ATET makes sense in job-training programs for which many types of individuals have zero chance of signing up. See also [Wooldridge \(2010, 911–913\)](#).

Second, the ATET reduces to the ATE when the mean of the covariates among the treated is the same as the mean of the covariates in the population and when the average contribution from the unobservables for the participants is zero.

Overview of treatment-effect estimators

We can classify the estimators implemented by `teffects` into five categories: 1) estimators based on a model for the outcome variable; 2) estimators based on a model for treatment assignment; 3) estimators based on models for both the outcome variable and the treatment assignment; 4) estimators that match on covariates; and 5) estimators that match on predicted probabilities of treatment. Within each category of estimator, there is a variety of choices about the functional forms for the models.

Because there are several categories of estimators, the user must decide whether to model the outcomes, the probability of treatment, or both. Ironically, some of the estimators that use both models only require that one of the two be correctly specified to consistently estimate the effects of interest, a property known as the double-robust property.

With the exception of using a matching estimator with a single continuous covariate, some choice of functional forms is required. There are two aspects one must consider when choosing the functional form for the outcome or treatment assignment. First, one must select the functional form for the conditional mean or conditional probability; depending on the variable being modeled, a linear, a binary choice, or an exponential model may be appropriate. Second, one must determine the appropriate polynomials of the covariates to include in the model. `teffects` offers a wide variety of options to specify different functional form choices for the conditional mean and conditional probability models. The factor variable notation in Stata allows us to easily specify the desired polynomial in the covariates.

We now provide some intuition behind each type of estimator.

RA estimators

RA estimators use means of predicted outcomes for each treatment level to estimate each POM. ATEs and ATETs are differences in estimated POMs.

The CI assumption implies that we can estimate $E(y_0|\mathbf{x})$ and $E(y_1|\mathbf{x})$ directly from the observations for which $t = 0$ and $t = 1$, respectively. Regression adjustment fits separate regressions for each treatment level and uses averages of the predicted outcomes over all the data to estimate the POMs. The estimated ATEs are differences in the estimated POMs. The estimated ATETs are averages of the predicted outcomes over the treated observations.

RA is a venerable estimator. See Lane and Nelder (1982); Cameron and Trivedi (2005, chap. 25); Wooldridge (2010, chap. 21); and Vittinghoff, Glidden, Shiboski, and McCulloch (2012, chap. 9). The usefulness of RA has been periodically questioned in the literature because it relies on specifying functional forms for the conditional means and because it requires having sufficient observations of each covariate pattern in each treatment level; see Rubin (1973) for an early salvo. Our experience is that RA is an exceptionally useful base-case estimator. We describe its relative advantages and disadvantages in the course of covering other estimators.

IPW estimators

IPW estimators use weighted averages of the observed outcome variable to estimate means of the potential outcomes. The weights account for the missing data inherent in the potential-outcome framework. Each weight is the inverse of the estimated probability that an individual receives a treatment level. Outcomes of individuals who receive a likely treatment get a weight close to one. Outcomes of individuals who receive an unlikely treatment get a weight larger than one, potentially much larger.

IPW estimators model the probability of treatment without any assumptions about the functional form for the outcome model. In contrast, RA estimators model the outcome without any assumptions about the functional form for the probability of treatment model.

IPW estimators become extremely unstable as the overlap assumption gets close to being violated. When the overlap assumption is nearly violated, some of the inverse-probability weights become very large, IPW estimators produce erratic estimates, and the large-sample distribution provides a poor approximation to the finite-sample distribution of IPW estimators. This instability occurs even though the functional form for the treatment model is correctly specified.

In contrast, when the overlap assumption is nearly violated, there are very few observations in a treatment level for some covariate patterns, so RA estimators use the model to predict in regions in which there are very little data. If the model is well specified and there are “enough” observations, an RA estimator will not become unstable as quickly as an IPW estimator, and the large-sample distribution of the RA estimator still provides a good approximation to the finite-sample distribution. However, in real situations in which “all models are approximate”, relying on a correctly specified outcome model with little data is extremely risky.

IPW estimators are a general approach to missing-data problems that obey some CI assumptions. While IPW is an old idea in statistics that dates back to Horvitz and Thompson (1952), biostatisticians and econometricians have been actively working on extending it to handle modern problems and estimation methods. See Robins and Rotnitzky (1995); Robins, Rotnitzky, and Zhao (1994, 1995); and Wooldridge (2002, 2007). IPW has been used extensively in the modern treatment-effect estimation literature. See Imbens (2000); Hirano, Imbens, and Ridder (2003); Tan (2010); Wooldridge (2010, chap. 19); van der Laan and Robins (2003); and Tsiatis (2006, chap. 6).

To see the intuition behind IPW, consider a study with observed outcome variable y , treatment variable $t \in \{0, 1\}$, and potential outcomes y_0 and y_1 . As part of this process, we need to estimate

the POM for treatment $t = 1$, $E(y_1)$. Using the observed data, $y_i t_i$ is y_{1i} when $t = 1$, but y_{1i} is unobserved when $t = 0$. An IPW estimator for $E(y_1)$ is $1/N \sum_{i=1}^N y_i t_i / p(\mathbf{x}_i)$, where $p(\mathbf{x}_i)$ is the probability that $t_i = 1$ and is a function of the covariates \mathbf{x}_i . If y_{1i} were always observed, the weights would all equal 1. This IPW estimator places a larger weight on those observations for which y_{1i} is observed even though its observation was not likely.

AIPW estimators

Instead of modeling either the outcome, like RA, or the treatment probability, like IPW, augmented inverse-probability-weighted (AIPW) estimators model both the outcome and the treatment probability. A surprising fact is that only one of the two models must be correctly specified to consistently estimate the treatment effects, a property of the AIPW estimators known as being “doubly robust”. Given that two models instead of one are used, it is less surprising that the AIPW estimators can be more efficient than either the RA or the IPW estimators, though deriving this result is rather technical and relies on the theory of semiparametric estimators.

Intuitively, the AIPW estimator is an IPW that includes an augmentation term that corrects the estimator when the treatment model is misspecified. When the treatment is correctly specified, the augmentation term vanishes as the sample size becomes large. Like the IPW, the AIPW does not perform well when the predicted treatment probabilities are too close to zero or one.

AIPW estimators emerge naturally from a technique of producing more efficient estimators from estimators that have a few main parameters of interest and some auxiliary, or nuisance, parameters used in estimating the few main parameters. This method constructs efficient estimating equations for the main parameters that are orthogonal to the auxiliary parameters. The estimators produced by this method are known as efficient-influence function (EIF) estimators.

To gain some intuition, consider finding an EIF estimator from an IPW estimator for two POMs. Note that we only care about the two POM parameters and not about the many auxiliary parameters used to estimate the treatment probabilities. EIF estimators project the equations that yield the POM parameters onto the equations that yield the auxiliary treatment-model parameters and then use the residuals from this projection to estimate the POM parameters.

We refer to these estimators as “AIPW estimators” instead of “EIF estimators” because the former is commonly used in the biostatistical literature for some of the binary-treatment estimators and because the term “augmented inverse-probability-weighted” tells more about how these estimators relate to the other implemented estimators; see [Tsiatis \(2006\)](#) and [Tan \(2010\)](#). The estimators implemented in `teffects aipw` with the `wls` option are based on those of [Rubin and van der Laan \(2008\)](#), which did well in simulations reported by [Tan \(2010\)](#), and denoted as $\tilde{\alpha}_{RV}(\hat{\pi})$ in [Tan \(2010, 663\)](#).

When either the outcome model or the treatment model is well specified, the AIPW estimators implemented in `teffects aipw` are more robust than either the RA or the IPW estimators because the AIPW estimators are doubly robust but the RA and IPW estimators are not. When both the outcome and the treatment model are misspecified, which estimator is more robust is a matter of debate in the literature; see [Kang and Schafer \(2007\)](#) and [Robins et al. \(2007\)](#) for some debate, and see [Tan \(2010\)](#) for a more recent discussion.

To the best of our knowledge, there is no general solution to the question of which estimator performs best when both the outcome and the treatment models are misspecified. We suspect that the answer depends on the true models, the implemented specifications, and the polynomials in the covariates used. To help users through this process, the estimators implemented in `teffects` offer many functional forms to approximate either the outcome process or the treatment process. In addition, Stata’s factor-variable notation makes it easy to include polynomials in the covariates. Both of these approximation methods rely on having enough data. `teffects` also makes it easy to compare the results produced by different estimators.

The literature on these methods is vast and growing. For double-robust results and explanations, see [Robins and Rotnitzky \(1995\)](#); [Robins, Rotnitzky, and Zhao \(1995\)](#); [van der Laan and Robins \(2003, chap. 6\)](#); [Bang and Robins \(2005\)](#); [Tsiatis \(2006, chap. 13\)](#); [Wooldridge \(2007; 2010, sec. 21.3.4\)](#); and [Tan \(2010\)](#).

IPWRA estimators

Like AIPW estimators, inverse-probability-weighted regression-adjustment (IPWRA) estimators combine models for the outcome and treatment status; also like AIPW estimators, IPWRA estimators are doubly robust. IPWRA estimators emerge naturally from a robust approach to missing-data methods. IPWRA estimators use the inverse of the estimated treatment-probability weights to estimate missing-data-corrected regression coefficients that are subsequently used to compute the POMs.

As far as we know, there is no literature that compares the relative efficiency of AIPW estimators, which emerge from a general approach to creating efficient estimators, and the IPWRA estimators, which emerge from a robust-correction approach to missing-data analysis.

The IPWRA estimators are also known as “Wooldridge’s double-robust” estimators because they were derived in [Wooldridge \(2007\)](#) and discussed at length in [Wooldridge \(2010, section 21.3.4\)](#).

Nearest-neighbor matching estimators

Matching estimators use an average of the outcomes of the nearest individuals to impute the missing potential outcome for each sampled individual. The difference between the observed outcome and the imputed potential outcome is an estimate of the individual-level treatment effect. These estimated individual-level treatment effects are averaged to estimate the ATE or the ATET.

`teffects nnmatch` determines the “nearest” by using a weighted function of the covariates for each observation. This type of matching is known as nearest-neighbor matching (NNM). `teffects psmatch` determines the “nearest” by using the estimated treatment probabilities, which are known as the propensity scores. This second type of matching is known as propensity-score matching (PSM).

NNM is nonparametric in that no explicit functional form for either the outcome model or the treatment model is specified. This flexibility comes at a price; the estimator needs more data to get to the true value than an estimator that imposes a functional form. More formally, the NNM estimator converges to the true value at a rate slower than the parametric rate, which is the square root of the sample size, when matching on more than one continuous covariate. `teffects nnmatch` uses bias correction to fix this problem. PSM provides an alternative to bias correction because it matches on a single continuous covariate, the estimated treatment probabilities.

[Abadie and Imbens \(2006, 2011\)](#) derived the rate of convergence of the NNM estimator and the bias-corrected NNM estimator and the large-sample distributions of the NNM and the bias-corrected NNM estimators. These articles provided the formal results that built on methods suggested in [Rubin \(1973, 1977\)](#).

`teffects nnmatch` is based on the results in [Abadie and Imbens \(2006, 2011\)](#) and a previous implementation in [Abadie et al. \(2004\)](#).

Propensity-score matching estimators

Instead of performing bias correction to handle the case of more than one continuous covariate, a common solution is to combine all the covariate information into estimated treatment probabilities, known as propensity scores, and use this single continuous covariate as the matching variable.

The term “propensity score” is widely used, but we continue to refer to it as the “treatment probability” to be consistent with the other estimators. We call the estimator that matches on the estimated treatment probabilities the “propensity-score matching (PSM) estimator” because the latter term is ubiquitous.

In effect, the PSM estimator parameterizes the bias-correction term in the treatment probability model. One advantage of matching on the estimated treatment probabilities over the bias-correction method is that one can explore the fit of different treatment probability models using standard methods before performing the nonparametric matching. For example, one can select the treatment model by minimizing an information criterion under i.i.d. sampling. We know of no counterpart for selecting the proper order of the bias-correction term for the NNM estimator.

Matching on estimated treatment probability models has been very popular since [Rosenbaum and Rubin \(1983\)](#) showed that if adjusting for covariates \mathbf{x}_i is sufficient to estimate the effects, then one can use the probability of treatment to perform the adjustment. [Abadie and Imbens \(2012\)](#) derived a method to estimate the standard errors of the estimator that matches on estimated treatment probabilities, and this method is implemented in `teffects psmatch`.

Choosing among estimators

There is no definitive way to select one of the estimators implemented in `teffects` over the others. We offer three observations about the tradeoffs among the estimators.

First, if the outcome model is correctly specified, the RA estimator will break down more slowly than the IPW, AIPW, IPWRA, or PSM estimators as the overlap assumption begins to fail. This result depends critically on the ability of the RA estimator to predict into regions in which there are little data.

Second, if the overlap assumption holds, the AIPW and IPWRA estimators have the double-robust property for some functional form combinations. The double-robust property says that if either the outcome model or the treatment model is correctly specified, we can consistently estimate the effects. The properties of double-robust estimators when both models are misspecified are not known, although there is some discussion in the literature about the properties of the AIPW estimators; see [Kang and Schafer \(2007\)](#), [Robins et al. \(2007\)](#), and [Tan \(2010\)](#).

Third, all the estimators require the same assumptions, so if each is correctly specified, they should all produce similar results. Of course, just because they produce similar results does not mean that they are correctly specified; it is possible that they are just behaving similarly in response to some underlying problem.

Model choice

`teffects` offers a broad selection of functional form combinations so that you can choose a combination that fits your data. Picking a functional form that respects the values of the observed outcomes is usually best. Select `linear` for continuous outcomes over the real line; `logit`, `probit`, or `hetprobit` for binary outcomes; and `poisson` for counts or nonnegative outcomes.

For binary treatments, you can select among `logit`, `probit`, or `hetprobit` models. For multivalued treatments, only the multinomial logit is available to model the treatment probabilities.

Selecting a functional form of a given set of covariates is a famously difficult problem in statistics. In the treatment-effects context, Cattaneo, Drukker, and Holland (2013) found that model selection by minimizing an information criterion worked well. Cattaneo, Drukker, and Holland (2013) discuss a method and a user-written command to facilitate the process.

Acknowledgments

We thank Matias Cattaneo, Guido Imbens, Barbara Sianesi, and Jeffrey Wooldridge for helpful conversations and advice about the literature and methods in this area.

We thank Matias Cattaneo for his collaboration on Cattaneo, Drukker, and Holland (2013) and for many conversations about Cattaneo (2010), a part of which is implemented in `teffects ipw` and `teffects aipw`.

We thank Ashley Holland for her collaboration on Cattaneo, Drukker, and Holland (2013).

We thank Alberto Abadie, Jane Herr, and Guido Imbens for collaboration on Abadie et al. (2004), which is implemented in `teffects nnmatch`.

We thank Matias Cattaneo for sharing an extract of the dataset that he used in Cattaneo (2010) and Barbara Sianesi for sharing an extract of the dataset that she used in Blundell, Dearden, and Sianesi (2005).

We thank Alberto Abadie, Guido Imbens, and László Sándor for sharing their implementation of the standard errors derived in Abadie and Imbens (2012), which are implemented in `teffects psmatch`.

We thank Edwin Leuven and Barbara Sianesi for their inspirational command `psmatch2`, which computes several of the estimators in `teffects` and a few more that we have not implemented.

References

- Abadie, A., D. M. Drukker, J. L. Herr, and G. W. Imbens. 2004. Implementing matching estimators for average treatment effects in Stata. *Stata Journal* 4: 290–311.
- Abadie, A., and G. W. Imbens. 2006. Large sample properties of matching estimators for average treatment effects. *Econometrica* 74: 235–267.
- . 2011. Bias-corrected matching estimators for average treatment effects. *Journal of Business and Economic Statistics* 29: 1–11.
- . 2012. Matching on the estimated propensity score. Harvard University and National Bureau of Economic Research. <http://www.hks.harvard.edu/fs/aabadie/pscore.pdf>.
- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.
- Bang, H., and J. M. Robins. 2005. Doubly robust estimation in missing data and causal inference models. *Biometrics* 61: 962–972.
- Becker, S. O., and M. Caliendo. 2007. Sensitivity analysis for average treatment effects. *Stata Journal* 7: 71–83.
- Blundell, R., L. Dearden, and B. Sianesi. 2005. Evaluating the effect of education on earnings: Models, methods and results from the National Child Development Survey. *Journal of the Royal Statistical Society, Series A* 168: 473–512.
- Brown, G. K., and T. Mergoupis. 2011. Treatment interactions with nonexperimental data in Stata. *Stata Journal* 11: 545–555.
- 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.

- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.
- Daniel, R. M., B. L. De Stavola, and S. N. Cousens. 2011. `gformula`: Estimating causal effects in the presence of time-varying confounding or mediation using the `g-computation` formula. *Stata Journal* 11: 479–517.
- Heckman, J. 1997. Instrumental variables: A study of implicit behavioral assumptions used in making program evaluations. *Journal of Human Resources* 32: 441–462.
- Heckman, J., and S. Navarro-Lozano. 2004. Using matching, instrumental variables, and control functions to estimate economic choice models. *Review of Economics and Statistics* 86: 30–57.
- Hirano, K., G. W. Imbens, and G. Ridder. 2003. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica* 71: 1161–1189.
- Holland, P. W. 1986. Statistics and causal inference. *Journal of the American Statistical Association* 81: 945–960.
- Horvitz, D. G., and D. J. Thompson. 1952. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association* 47: 663–685.
- Imbens, G. W. 2000. The role of the propensity score in estimating dose-response functions. *Biometrika* 87: 706–710.
- . 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics* 86: 4–29.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86.
- Kang, J. D. Y., and J. L. Schafer. 2007. Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data. *Statistical Science* 22: 523–539.
- Lane, P. W., and J. A. Nelder. 1982. Analysis of covariance and standardization as instances of prediction. *Biometrics* 38: 613–621.
- Nichols, A. 2007. Causal inference with observational data. *Stata Journal* 7: 507–541.
- Robins, J. M. 1986. A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical Modeling* 7: 1393–1512.
- Robins, J. M., and A. Rotnitzky. 1995. Semiparametric efficiency in multivariate regression models with missing data. *Journal of the American Statistical Association* 90: 122–129.
- Robins, J. M., A. Rotnitzky, and L. P. Zhao. 1994. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association* 89: 846–866.
- . 1995. Analysis of semiparametric regression models for repeated outcomes in the presence of missing data. *Journal of the American Statistical Association* 90: 106–121.
- Robins, J. M., M. Sued, Q. Lei-Gomez, and A. Rotnitzky. 2007. Comment: Performance of double-robust estimators when “inverse probability” weights are highly variable. *Statistical Science* 22: 544–559.
- 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.
- Rubin, D. B. 1973. Matching to remove bias in observational studies. *Biometrics* 29: 159–183.
- . 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701.
- . 1977. Assignment to treatment group on the basis of a covariate. *Journal of Educational Statistics* 2: 1–26.
- Rubin, D. B., and M. J. van der Laan. 2008. Empirical efficiency maximization: Improved locally efficient covariate adjustment in randomized experiments and survival analysis. *International Journal of Biostatistics* 4: 1–40.
- Tan, Z. 2010. Bounded, efficient and doubly robust estimation with inverse weighting. *Biometrika* 97: 661–682.
- Tsiatis, A. A. 2006. *Semiparametric Theory and Missing Data*. New York: Springer.
- van der Laan, M. J., and J. M. Robins. 2003. *Unified Methods for Censored Longitudinal Data and Causality*. 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. 2002. Inverse probability weighted M-estimators for sample selection, attrition, and stratification. *Portuguese Economic Journal* 1: 117–139.

- . 2007. Inverse probability weighted estimation for general missing data problems. *Journal of Econometrics* 141: 1281–1301.
- . 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

[TE] **teffects** — Treatment-effects estimation for observational data

[TE] **teffects intro** — Introduction to treatment effects for observational data

Title

teffects aipw — Augmented inverse-probability weighting

Syntax

Remarks and examples

Also see

Menu

Stored results

Description

Methods and formulas

Options

References

Syntax

```
teffects aipw (ovar omvarlist [, omodel noconstant])  
  (tvar tmvarlist [, tmodel noconstant]) [if] [in] [weight]  
  [, stat options]
```

<i>omodel</i>	Description
---------------	-------------

Model

<code>linear</code>	linear outcome model; the default
<code>logit</code>	logistic outcome model
<code>probit</code>	probit outcome model
<code>hetprobit(<i>varlist</i>)</code>	heteroskedastic probit outcome model
<code>poisson</code>	exponential outcome model

omodel specifies the model for the outcome variable.

<i>tmodel</i>	Description
---------------	-------------

Model

<code>logit</code>	logistic treatment model; the default
<code>probit</code>	probit treatment model
<code>hetprobit(<i>varlist</i>)</code>	heteroskedastic probit treatment model

tmodel specifies the model for the treatment variable.

For multivariate treatments, only `logit` is available and multinomial `logit` is used.

<i>stat</i>	Description
-------------	-------------

Stat

<code>ate</code>	estimate average treatment effect in population; the default
<code>pomeans</code>	estimate potential-outcome means

<i>options</i>	Description
Model	
<code>nls</code>	estimate conditional means by nonlinear least squares
<code>wnls</code>	estimate conditional means by weighted nonlinear least squares
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>robust</code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>aequations</code>	display auxiliary-equation results
<code>display_options</code>	control column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
Advanced	
<code>pstolerance(#)</code>	set tolerance for overlap assumption
<code>osample(<i>newvar</i>)</code>	<i>newvar</i> identifies observations that violate the overlap assumption
<code>control(# <i>label</i>)</code>	specify the level of <i>tvar</i> that is the control
<code>coeflegend</code>	display legend instead of statistics

tvar must contain integer values representing the treatment levels.

omvarlist and *tmvarlist* may contain factor variables; see [U] 11.4.3 **Factor variables**.

`bootstrap`, `by`, `jackknife`, and `statsby` are allowed; see [U] 11.1.10 **Prefix commands**.

`fweights` and `iweights` are allowed; see [U] 11.1.6 **weight**.

`coeflegend` does not appear in the dialog box.

See [U] 20 **Estimation and postestimation commands** for more capabilities of estimation commands.

Menu

Statistics > Treatment effects > Doubly robust estimators > Augmented inverse-probability weights

Description

`teffects aipw` estimates treatment effects from observational data via augmented inverse-probability weighting. AIPW estimators combine aspects of regression-adjustment and inverse-probability-weighted methods to estimate potential-outcome means and average treatment effects. AIPW estimators possess the doubly robust property. `teffects aipw` offers several choices for the functional forms of the outcome model and of the treatment model.

See [TE] **teffects intro** or [TE] **teffects intro advanced** for more information about these estimators.

Options

Model

`noconstant`; see [R] **estimation options**.

`nls` specifies that the parameters of the outcome model be estimated by nonlinear least squares instead of the default maximum likelihood.

`wnls` specifies that the parameters of the outcome model be estimated by weighted nonlinear least squares instead of the default maximum likelihood. The weights make the estimator of the effect parameters more robust to a misspecified outcome model.

Stat

`stat` is one of two statistics: `ate` or `pomeans`. `ate` is the default.

`ate` specifies that the average treatment effect be estimated.

`pomeans` specifies that the potential-outcome means for each treatment level be estimated.

SE/Robust

`vce(vctype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`) and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce_option](#).

Reporting

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

`aequations` specifies that the results for the outcome-model or the treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

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

Maximization

`maximize_options`: `iterate(#)`, `[no]log`, and `from(init_specs)`; see [R] [maximize](#). These options are seldom used.

`init_specs` is one of

`matname` [, `skip copy`]

`#` [, `# ...`] , `copy`

Advanced

`pstolerance(#)` specifies the tolerance used to check the overlap assumption. The default value is `pstolerance(1e-5)`. `teffects` will exit with an error if an observation has an estimated propensity score smaller than that specified by `pstolerance()`.

`osample(newvar)` specifies that indicator variable `newvar` be created to identify observations that violate the overlap assumption.

`control(# | label)` specifies the level of `tvar` that is the control. The default is the first treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with statistic `pomeans`.

The following option is available with `teffects aipw` but is not shown in the dialog box:

`coeflegend`; see [R] [estimation options](#).

Remarks and examples

Remarks are presented under the following headings:

[Overview](#)

[Video example](#)

Overview

AIPW estimators use inverse-probability weights to correct for the missing-data problem arising from the fact that each subject is observed in only one of the potential outcomes; these estimators also use an augmentation term in the outcome model to correct the estimator in case the treatment model is misspecified. If the treatment model is correctly specified, the augmentation term goes to zero in large samples.

AIPW estimators compute averages of the augmented inverse-probability-weighted outcomes for each treatment level. Contrasts of these averages provide estimates of the treatment effects.

AIPW estimators use a model to predict treatment status, and they use another model to predict outcomes. Because of the double-robust property, only one of these two models must be correctly specified for the AIPW estimator to be consistent.

AIPW estimators use a three-step approach to estimating treatment effects:

1. They estimate the parameters of the treatment model and compute inverse-probability weights.
2. They estimate separate regression models of the outcome for each treatment level and obtain the treatment-specific predicted outcomes for each subject.
3. They compute the weighted means of the treatment-specific predicted outcomes, where the weights are the inverse-probability weights computed in step 1. The contrasts of these weighted averages provide the estimates of the ATEs.

These steps produce consistent estimates of the effect parameters because the treatment is assumed to be independent of the potential outcomes after conditioning on the covariates. The overlap assumption ensures that predicted inverse-probability weights do not get too large. The standard errors reported by `teffects aipw` correct for the three-step process. See [TE] [teffects intro](#) or [TE] [teffects intro advanced](#) for more information about this estimator.

We will illustrate the use of `teffects aipw` by using data from a study of the effect of a mother's smoking status during pregnancy (`mbsmoke`) on infant birthweight (`bweight`) as reported by [Cattaneo \(2010\)](#). This dataset also contains information about each mother's age (`mage`), education level (`medu`), marital status (`mmarried`), whether the first prenatal exam occurred in the first trimester (`prenatal1`), and whether this baby was the mother's first birth (`fbaby`).

▷ Example 1: Estimating the ATE

We begin by using `teffects aipw` to estimate the average treatment effect of `mbsmoke` on `bweight`. We use a probit model to predict treatment status as a function of `mmarried`, `mage`, and `fbaby`; to maximize the predictive power of this model, we use factor-variable notation to incorporate quadratic effects of the mother's age, the only continuous covariate in our model. We use linear regression to model birthweight, using `prenatal1`, `mmarried`, `mage`, and `fbaby` as explanatory variables. We type

```

. use http://www.stata-press.com/data/r13/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects aipw (bweight prenatal1 mmarried mage fbaby)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, probit)

Iteration 0:  EE criterion = 4.622e-21
Iteration 1:  EE criterion = 1.801e-25

Treatment-effects estimation          Number of obs      =      4642
Estimator      : augmented IPW
Outcome model  : linear by ML
Treatment model: probit

```

bweight	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
ATE						
mbsmoke (smoker vs nonsmoker)	-230.9892	26.21056	-8.81	0.000	-282.361	-179.6174
POMean						
mbsmoke nonsmoker	3403.355	9.568472	355.68	0.000	3384.601	3422.109

Smoking causes infants' birthweights to be reduced by an average of 231 grams from the average of 3,403 grams for infants of mothers who do not smoke.

◀

By default, `teffects aipw` reports the ATE and the POM for the base (untreated) subjects. The `pomeans` option allows us to view the treated subjects' POM as well; the `aequations` option displays the regression model coefficients used to predict the POMs as well as the coefficients from the model used to predict treatment.

► Example 2: Displaying the POMs and equations

Here we use the `pomeans` and `aequations` options to obtain estimates of both POMs and view all the fitted equations underlying our estimates:

```
. teffects aipw (bweight prenatal1 mmarried mage fbaby)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, probit), pomeans aequations
Iteration 0: EE criterion = 4.622e-21
Iteration 1: EE criterion = 9.572e-26
Treatment-effects estimation                               Number of obs      =      4642
Estimator          : augmented IPW
Outcome model      : linear by ML
Treatment model:    probit
```

bweight	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
POmeans						
mbsmoke						
nonsmoker	3403.355	9.568472	355.68	0.000	3384.601	3422.109
smoker	3172.366	24.42456	129.88	0.000	3124.495	3220.237
OME0						
prenatal1	64.40859	27.52699	2.34	0.019	10.45669	118.3605
mmarried	160.9513	26.6162	6.05	0.000	108.7845	213.1181
mage	2.546828	2.084324	1.22	0.222	-1.538373	6.632028
fbaby	-71.3286	19.64701	-3.63	0.000	-109.836	-32.82117
_cons	3202.746	54.01082	59.30	0.000	3096.886	3308.605
OME1						
prenatal1	25.11133	40.37541	0.62	0.534	-54.02302	104.2457
mmarried	133.6617	40.86443	3.27	0.001	53.5689	213.7545
mage	-7.370881	4.21817	-1.75	0.081	-15.63834	.8965804
fbaby	41.43991	39.70712	1.04	0.297	-36.38461	119.2644
_cons	3227.169	104.4059	30.91	0.000	3022.537	3431.801
TME1						
mmarried	-.6484821	.0554173	-11.70	0.000	-.757098	-.5398663
mage	.1744327	.0363718	4.80	0.000	.1031452	.2457202
c.mage#						
c.mage	-.0032559	.0006678	-4.88	0.000	-.0045647	-.0019471
fbaby	-.2175962	.0495604	-4.39	0.000	-.3147328	-.1204595
medu	-.0863631	.0100148	-8.62	0.000	-.1059917	-.0667345
_cons	-1.558255	.4639691	-3.36	0.001	-2.467618	-.6488926

The coefficient table indicates that the treated POM is 3,172 grams, 231 grams less than the untreated POM. The sections of the table labeled OME0 and OME1 represent the linear regression coefficients for the untreated and treated potential-outcome equations, respectively. The coefficients of the TME1 equation are used in the probit model to predict treatment status.

◀

As is well known, the standard probit model assumes that the error terms in the latent-utility framework are homoskedastic; the model is not robust to departures from this assumption. An alternative is to use the heteroskedastic probit model, which explicitly models the error variance as a function of a set of variables.

▶ Example 3: Heteroskedastic probit treatment model

Here we refit our model as in the previous examples, but we instead use heteroskedastic probit to model the treatment variable. We posit that the heteroskedasticity is a function of the mother's age. We type

```

. teffects aipw (bweight prenatal1 mmarried fbaby)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, hetprobit(c.mage)), aequations
Iteration 0: EE criterion = 1.746e-19
Iteration 1: EE criterion = 1.746e-19 (backed up)
Treatment-effects estimation          Number of obs      =      4642
Estimator      : augmented IPW
Outcome model  : linear by ML
Treatment model: heteroskedastic probit

```

bweight	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
ATE						
mbsmoke (smoker vs nonsmoker)	-230.2699	27.49461	-8.38	0.000	-284.1584	-176.3815
POmean						
mbsmoke nonsmoker	3403.657	9.540713	356.75	0.000	3384.957	3422.356
OME0						
prenatal1	69.5048	27.04642	2.57	0.010	16.49479	122.5148
mmarried	173.74	24.63865	7.05	0.000	125.4491	222.0308
fbaby	-79.19473	18.62584	-4.25	0.000	-115.7007	-42.68875
_cons	3260.768	28.29282	115.25	0.000	3205.315	3316.221
OME1						
prenatal1	12.86437	39.83916	0.32	0.747	-65.21894	90.94768
mmarried	113.3491	39.47422	2.87	0.004	35.9811	190.7172
fbaby	64.22326	38.42042	1.67	0.095	-11.07939	139.5259
_cons	3051.268	37.30413	81.79	0.000	2978.153	3124.383
TME1						
mmarried	-.3551755	.1044199	-3.40	0.001	-.5598347	-.1505163
mage	.0831898	.0349088	2.38	0.017	.0147699	.1516097
c.mage# c.mage	-.0013458	.0006659	-2.02	0.043	-.002651	-.0000406
fbaby	-.1170697	.044998	-2.60	0.009	-.2052643	-.0288752
medu	-.0435057	.0147852	-2.94	0.003	-.0724842	-.0145272
_cons	-.8757021	.3478139	-2.52	0.012	-1.557405	-.1939994
TME1_lnsigma						
mage	-.0236336	.0107134	-2.21	0.027	-.0446315	-.0026357

The equation labeled TME1_lnsigma represents the heteroskedasticity function used to model the logarithm of the variance. Because the coefficient on the single variable we specified is significant below the 5% level, we conclude that allowing for heteroskedasticity was indeed necessary.

◀

Rather than using maximum likelihood to fit the outcome model, you can instruct `teffects aipw` to use a weighted nonlinear least-squares (WNLS) estimator instead. The WNLS estimator may be more robust to outcome model misspecification.

▶ Example 4: Using the WNLS estimator

Here we use WNLS to fit our outcome model:

```
. use http://www.stata-press.com/data/r13/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects aipw (bweight prenatal1 mmarried mage fbaby)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, probit), wnls
Iteration 0:  EE criterion = 1.963e-20
Iteration 1:  EE criterion = 8.897e-25
Treatment-effects estimation      Number of obs      =      4642
Estimator      : augmented IPW
Outcome model  : linear by WNLS
Treatment model: probit
```

bweight	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
ATE mbsmoke (smoker vs nonsmoker)	-227.1956	27.34793	-8.31	0.000	-280.7966	-173.5947
POmean mbsmoke nonsmoker	3403.251	9.596622	354.63	0.000	3384.442	3422.06

◀

Video example

Treatment effects: Augmented inverse probability weights

Stored results

`teffects aipw` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level j
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	<code>teffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(subcmd)</code>	<code>aipw</code>
<code>e(tmodel)</code>	logit, probit, or hetprobit
<code>e(omodel)</code>	linear, logit, probit, hetprobit, or poisson
<code>e(stat)</code>	statistic estimated, ate or pomeans
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(tlevels)</code>	levels of treatment variable
<code>e(cme)</code>	ml, nls, or wnl
<code>e(vce)</code>	<i>vctype</i> specified in <code>vce()</code>
<code>e(vctype)</code>	title used to label Std. Err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

Methods and formulas

The methods and formulas presented here provide the technical details underlying the estimators implemented in `teffects ra`, `teffects ipw`, `teffects aipw`, and `teffects ipwra`. See [Methods and formulas](#) of [TE] `teffects nnmatch` for the methods and formulas used by `teffects nnmatch` and `teffects psmatch`.

Methods and formulas are presented under the following headings:

Parameters and notation

Overview of EE estimators

VCE for EE estimators

TM and OM estimating functions

TM estimating functions

OM estimating functions

Effect estimating functions

RA estimators

IPW estimators

AIPW estimators

IPWRA estimators

Parameters and notation

We begin by reviewing the effect parameters estimated by `teffects` and some essential notation.

The potential outcome that an individual would obtain if given treatment level $t \in \{0, 1, \dots, q\}$ is y_t . Each y_t is a random variable, the realizations of which are y_{ti} . Throughout this document, i subscripts denote realizations of the corresponding, unsubscripted random variables.

The three parameters of interest are

1. the potential-outcome mean (POM) $\alpha_t = E(y_t)$;
2. the average treatment effect (ATE) $\tau_t = E(y_t - y_0)$; and
3. the average treatment effect on the treated (ATET) $\delta_t = E(y_t - y_0 | t = \tilde{t})$.

The no-treatment level is 0.

The estimators implemented in `teffects` use three assumptions to justify the equations used for estimation and inference about the effect parameters of interest:

1. Conditional mean independence (CMI) allows us to estimate potential-outcome means from the observed outcomes in the sample.
2. Overlap ensures that we have data on each type of individual in each treatment level.
3. Independent observations ensure that the outcome and treatment for one individual has no effect on the outcome or treatment for any other individual.

`teffects ra` implements some regression-adjustment (RA) estimators; `teffects ipw` implements some inverse-probability-weighted (IPW) estimators; `teffects ipwra` implements some inverse-probability-weighted regression-adjustment (IPWRA) estimators; and `teffects aipw` implements some augmented inverse-probability-weighted (AIPW) estimators. All are implemented as estimating-equation (EE) estimators. The estimators are consistent and asymptotically normally distributed under the CMI, overlap, and independence assumptions.

Overview of EE estimators

EE estimators compute estimates by solving sample estimating equations. The sample estimating equations are the sample equivalents of population expectation equations.

Each EE estimator specifies a set of estimating equations for the effect parameters of interest and a set of estimating equations for the auxiliary parameters in the outcome model (OM) or the treatment model (TM). The next few sections provide tremendous detail about the estimating equations that define the RA, IPW, AIPW, and IPWRA estimators.

Ignoring the details for a moment, EE estimators solve systems of equations to compute estimates. A standard robust estimator is consistent for the variance of the estimator (VCE). All the details involve the equations specified by choices of estimator and functional forms for the OM or TM.

When used, the OM is a model for the conditional mean of the outcome variable. We let $\mu(\mathbf{x}, t, \beta_t)$ denote a conditional-mean model for the outcome y conditional on covariates \mathbf{x} and treatment level t . Mathematically, $E(y|\mathbf{x}, t) = \mu(\mathbf{x}, t, \beta_t)$, where β_t are the parameters of the conditional-mean model given treatment level t . The table below provides details about the available functional forms.

Outcome model	Functional form for $\mu(\mathbf{x}, t, \beta_t)$
linear	$\mathbf{x}\beta_t$
logit	$\exp(\mathbf{x}\beta_t)/\{1 + \exp(\mathbf{x}\beta_t)\}$
probit	$\Phi(\mathbf{x}\beta_t)$
poisson	$\exp(\mathbf{x}\beta_t)$
hetprobit	$\Phi(\dot{\mathbf{x}}\dot{\beta}_t/\ddot{\mathbf{x}}\ddot{\beta}_t)$

In the case of `hetprobit`, we use $\dot{\mathbf{x}}$ and $\dot{\beta}_t$ to denote the variables and parameters in the index function, and we use $\ddot{\mathbf{x}}$ and $\ddot{\beta}_t$ to denote the variables and parameters in the variance equation. We define $\mathbf{x}' = (\dot{\mathbf{x}}', \ddot{\mathbf{x}}')$ and $\beta'_t = (\dot{\beta}'_t, \ddot{\beta}'_t)$.

We write the vector of parameters for the outcome model over all treatment levels as $\beta' = (\beta'_0, \beta'_1, \dots, \beta'_q)$.

Next we provide details about the estimating equations implied by each functional form choice.

When used, the TM is a model for the conditional probability of treatment. We let $p(\mathbf{z}, t, \gamma)$ denote the conditional probability model for the probability that a person receives treatment t , conditional on covariates \mathbf{z} . The table below provides details about the functional form options allowed in the case of a binary treatment.

Treatment model	Functional form for $p(\mathbf{z}, t, \gamma)$
logit	$\exp(\mathbf{z}\gamma)/\{1 + \exp(\mathbf{z}\gamma)\}$
probit	$\Phi(\mathbf{z}\gamma)$
hetprobit	$\Phi(\dot{\mathbf{z}}\dot{\gamma}/\ddot{\mathbf{z}}\ddot{\gamma})$

In the case of `hetprobit`, we use $\dot{\mathbf{z}}$ and $\dot{\gamma}$ to denote the variables and parameters in the index function, and we use $\ddot{\mathbf{z}}$ and $\ddot{\gamma}$ to represent the variables and parameters in the variance equation. We define $\mathbf{z}' = (\dot{\mathbf{z}}', \ddot{\mathbf{z}}')$, and $\gamma' = (\dot{\gamma}', \ddot{\gamma}')$.

In the multivalued-treatment case, $p(\mathbf{z}, t, \gamma)$ is specified as a multinomial logit with $p(\mathbf{z}, t, \gamma) = \exp(\mathbf{z}\gamma_t)/\{1 + \sum_{k=1}^q \exp(\mathbf{z}\gamma_k)\}$ and $\gamma' = (\gamma'_1, \gamma'_2, \dots, \gamma'_q)$. (We present formulas for the case with treatment level 0 as the base with $\gamma'_0 = \mathbf{0}'$; see [R] `mlogit` for background.) In `teffects`, the `logit` option in the treatment-model specification means binary logit for the binary-treatment case and multinomial logit for the multivalued-treatment case: this simplifies the use of the command and makes statistical sense.

Below we provide details about the estimating equations implied by each functional form choice. The effect parameters of interest are

1. the POMs denoted by $\alpha' = (\alpha_0, \alpha_1, \dots, \alpha_q)$;
2. the ATEs denoted by $\tau' = (\tau_1, \tau_2, \dots, \tau_q)$; and
3. the ATETs denoted by $\delta' = (\delta_1, \delta_2, \dots, \delta_q)$.

We denote the effect parameters by ϑ and all the parameters in any particular case by θ . More formally, θ is the concatenation of the effect parameters, the OM parameters, and the TM parameters; $\theta' = (\vartheta', \beta', \gamma')$, where ϑ is α , τ , or δ , and β or γ may not be present, depending on the case at hand.

In the subsections below, we discuss estimators for the elements in θ in detail and note how these elements change over the cases defined by effect parameters and estimators. The parameter vector θ denotes all the parameters, no matter which particular case is under consideration.

The EE estimators described in this section are defined by a set of equations,

$$E\{\mathbf{s}(\mathbf{x}, \mathbf{z}, \theta)\} = \mathbf{0}$$

where $\mathbf{s}(\mathbf{x}, \mathbf{z}, \theta)$ is a vector of estimating functions. Note the notation: estimating equations equate the expected value of a vector of estimating functions to zero.

Because each of the estimating functions has mean zero, we can construct estimators that find the estimates $\hat{\theta}$ by solving a system of equations,

$$1/N \sum_i^N \mathbf{s}_i(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta}) = \mathbf{0}$$

where $\mathbf{s}_i(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta})$ are the sample realizations of the estimating functions. In other words, the parameter estimates set the average of the realizations of each estimating function to zero. Almost all the details below involve specifying the sample realizations $\mathbf{s}_i(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta})$.

Estimators that set the expected value of estimating functions to zero are known as estimating-equations (EE) estimators, M estimators, or Z estimators in the statistics literature and as generalized method of moments (GMM) estimators in the econometrics literature. See [van der Vaart \(1998, 41\)](#), [Stefanski and Boos \(2002\)](#), and [Tsiatis \(2006, sec. 3.2\)](#) for statistics; and see [Wooldridge \(2010, chap. 14\)](#), [Cameron and Trivedi \(2005, chap. 6\)](#), and [Newey and McFadden \(1994\)](#) for econometrics.

We refer to them as EE estimators because this name is closest to being independent of any discipline. The estimators are implemented using `gmm` because they are exactly identified generalized method-of-moments (GMM) estimators. When weights are specified by the user, they are applied to the estimating equations just as `gmm` applies user-specified weights.

Each estimator has a set of estimating equations for the effect parameters and either an OM or a TM, or both. The OM parameters or the TM parameters are auxiliary parameters used to estimate the effect parameters of interest.

Each set of parameters has its own set of sample estimating equations:

$1/N \sum_i \mathbf{s}_{e,i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta}) = \mathbf{0}$ are the sample estimating equations for the effect parameters. These equations determine the effect parameter estimates $\hat{\theta}$ as functions of the data and the other estimated parameters.

$1/N \sum_i \mathbf{s}_{\text{om},i}(\mathbf{x}_i, w_i, \hat{\beta}) = \mathbf{0}$ are the sample estimating equations for OM parameters that use the weights w_i , which are functions of the TM parameters.

$1/N \sum_i \mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \hat{\gamma}) = \mathbf{0}$ are the sample estimating equations for TM parameters.

The whole set of sample estimating functions is $\mathbf{s}_i(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta})$ with

$$\mathbf{s}_i(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta})' = (\mathbf{s}_{e,i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta})', \mathbf{s}_{\text{om},i}(\mathbf{x}_i, w_i(t), \hat{\beta})', \mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \hat{\gamma})')$$

although not all the estimators have each of three components.

VCE for EE estimators

The Huber/White/robust sandwich estimator is consistent for the variance–covariance of the estimator (VCE). See [van der Vaart \(1998, 41\)](#), [Stefanski and Boos \(2002\)](#), and [Tsiatis \(2006, sec. 3.2\)](#) for statistics; and see [Wooldridge \(2010, chap. 14\)](#), [Cameron and Trivedi \(2005, chap. 6\)](#), and [Newey and McFadden \(1994\)](#) for econometrics.

The formula is

$$\widehat{\mathbf{V}} = (1/N)\overline{\mathbf{G}} \overline{\mathbf{S}} \overline{\mathbf{G}}'$$

where

$$\overline{\mathbf{G}} = \left\{ (1/N) \sum_i \frac{\partial s_i(\mathbf{x}_i, \mathbf{z}_i, \widehat{\boldsymbol{\theta}})}{\partial \widehat{\boldsymbol{\theta}}} \right\}^{-1}$$

and

$$\overline{\mathbf{S}} = (1/N) \sum_i s_i(\mathbf{x}_i, \mathbf{z}_i, \widehat{\boldsymbol{\theta}}) s_i(\mathbf{x}_i, \mathbf{z}_i, \widehat{\boldsymbol{\theta}})'$$

The matrix $\overline{\mathbf{G}}$ is not symmetric because our EE estimators come from stacking moment conditions instead of optimizing a single objective function. The implication is that the robust formula should always be used because, even under correct specification, the nonsymmetric $\overline{\mathbf{G}}$ and the symmetric $\overline{\mathbf{S}}$ converge to different matrices.

TM and OM estimating functions

Although the sample estimating functions for the effect parameters, the $s_{e,i}(\mathbf{x}_i, \mathbf{z}_i, \widehat{\boldsymbol{\theta}})$, are estimator specific, the sample estimating functions for the TM parameters, the $s_{\text{tm},i}(\mathbf{z}_i, \widehat{\boldsymbol{\gamma}})$, and the sample estimating functions for the OM parameters, the $s_{\text{om},i}(\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}})'$, are used in multiple estimators. We provide details about the TM and the OM sample estimating functions here.

TM estimating functions

The sample estimating functions used to estimate the parameters of the TM $p(\mathbf{z}, t, \boldsymbol{\gamma})$ are the sample score equations from the quasimaximum likelihood (QML) estimator.

In the binary-treatment case, $p(\mathbf{z}, t, \boldsymbol{\gamma})$ may be logit, probit, or heteroskedastic probit. In the multivalued-treatment case, $p(\mathbf{z}, t, \boldsymbol{\gamma})$ is a multinomial logit. We now give formulas for the $s_{\text{tm},i}(\mathbf{z}_i, \widehat{\boldsymbol{\gamma}})$ for each case.

logit and probit

In the logit and probit cases,

$$s_{\text{tm},i}(\mathbf{z}_i, \widehat{\boldsymbol{\gamma}}) = \left[\frac{g(\mathbf{z}_i \widehat{\boldsymbol{\gamma}}') \{t_i - G(\mathbf{z}_i \widehat{\boldsymbol{\gamma}}')\}}{G(\mathbf{z}_i \widehat{\boldsymbol{\gamma}}') \{1 - G(\mathbf{z}_i \widehat{\boldsymbol{\gamma}}')\}} \right] \mathbf{z}_i$$

where $G(z)$ is the logistic cumulative distribution function for the logit, $G(z)$ is the normal cumulative distribution function for the probit, and $g(\cdot) = \{\partial G(z)\}/(\partial z)$ is the corresponding density function.

hetprobit

In the `hetprobit` case, there are two sets of sample score equations, $\mathbf{s}_{\text{tm},1,i}(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})$ and $\mathbf{s}_{\text{tm},2,i}(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})$:

$$\mathbf{s}_{\text{tm},1,i}(\mathbf{z}_i, \hat{\boldsymbol{\gamma}}) = \left(\frac{\phi \{q(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})\} [t_i - \Phi \{q(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})\}]}{\Phi \{q(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})\} [1 - \Phi \{q(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})\}] \exp(\ddot{\mathbf{z}}_i \hat{\boldsymbol{\gamma}})} \right) \dot{\mathbf{z}}_i'$$

and

$$\mathbf{s}_{\text{tm},2,i}(\mathbf{z}_i, \hat{\boldsymbol{\gamma}}) = \left(\frac{\phi \{q(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})\} \dot{\mathbf{z}}_i \hat{\boldsymbol{\gamma}}' [\Phi \{q(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})\} - t_i]}{\Phi \{q(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})\} [1 - \Phi \{q(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})\}] \exp(\ddot{\mathbf{z}}_i \hat{\boldsymbol{\gamma}})} \right) \ddot{\mathbf{z}}_i'$$

where $\phi(\cdot)$ is the standard normal density function.

mlogit

In the `mlogit` case, $p(\mathbf{z}, t, \boldsymbol{\gamma}) = \exp(\mathbf{z}\boldsymbol{\gamma}_t) / \{1 + \sum_{k=1}^q \exp(\mathbf{z}\boldsymbol{\gamma}_k)\}$. We present formulas for the case with treatment level 0 as the base with $\boldsymbol{\gamma}'_0 = \mathbf{0}'$; see [R] `mlogit` for background.

There are q vectors of sample estimating functions for the `mlogit` case, $\mathbf{s}_{\text{tm},k,i}(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})$ for each $k \in \{1, \dots, q\}$, 1 for each vector $\hat{\boldsymbol{\gamma}}_k$, $k \in \{1, \dots, q\}$. These sample estimating functions are

$$\mathbf{s}_{\text{tm},k,i}(\mathbf{z}_i, \hat{\boldsymbol{\gamma}}) = \begin{cases} \{1 - p(\mathbf{z}_i, k, \hat{\boldsymbol{\gamma}})\} \mathbf{z}'_i & t_i = k \\ -p(\mathbf{z}_i, k, \hat{\boldsymbol{\gamma}}) \mathbf{z}'_i & \text{otherwise} \end{cases}$$

OM estimating functions

The parameters of the OM $\mu(\mathbf{x}, t, \boldsymbol{\beta}_t)$ are estimated either by weighted QML or by weighted nonlinear least squares. The estimating functions used to estimate the parameters of the OM are either the score equations from the weighted QML estimator or the moment conditions for the weighted nonlinear least-squares estimator.

The estimating functions for the OM parameters in $\mu(\mathbf{x}, t, \boldsymbol{\beta}_t)$ vary over the models for the conditional mean because $\mu(\mathbf{x}, t, \boldsymbol{\beta}_t)$ may be linear, logit, probit, heteroskedastic probit, or poisson.

Let N_t be the number of observations in treatment level t , and let $t_i(t) = 1$ if $t_i = t$, with $t_i(t) = 0$ if $t_i \neq t$.

There are two sets of sample estimating functions for the OM parameters with weights $w_i(t)$:

1. $\mathbf{s}_{\text{ml,om},i}\{\mathbf{x}_i, w_i(t), \hat{\boldsymbol{\beta}}_t\}$ are the sample estimating functions for the weighted QML estimator.
2. $\mathbf{s}_{\text{nls,om},i}\{\mathbf{x}_i, w_i(t), \hat{\boldsymbol{\beta}}_t\}$ are the sample estimating functions for the weighted nonlinear least-squares estimator.

OM QML

Here are the formulas for the $\mathbf{s}_{\text{ml,om},i}\{\mathbf{x}_i, w_i(t), \hat{\boldsymbol{\beta}}_t\}$ for each functional form choice.

linear

In the linear case,

$$\mathbf{s}_{\text{ml,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t)(y_i - \mathbf{x}_i\widehat{\boldsymbol{\beta}}_t)\mathbf{x}_i'$$

logit and probit

In the logit and probit cases,

$$\mathbf{s}_{\text{ml,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t) \left[\frac{g(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t') \{y_i - G(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t')\}}{G(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t') \{1 - G(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t')\}} \right] \mathbf{x}_i$$

where $G(z)$ is the logistic cumulative distribution function for the logit, $G(z)$ is the normal cumulative distribution function for the probit, and $g(\cdot) = \{\partial G(z)\}/(\partial z)$ is the corresponding density function.

hetprobit

In the **hetprobit** case, there are two sets of sample score equations, $\mathbf{s}_{\text{ml,om},1,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}$ and $\mathbf{s}_{\text{ml,om},2,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}$:

$$\mathbf{s}_{\text{ml,om},1,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t) \left(\frac{\phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\} [y_i - \Phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\}]}{\Phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\} [1 - \Phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\}]} \exp(\ddot{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}_t) \right) \ddot{\mathbf{x}}_i'$$

and

$$\mathbf{s}_{\text{ml,om},2,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t) \left(\frac{\phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\} \ddot{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}_t' [\Phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\} - y_i]}{\Phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\} [1 - \Phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\}]} \exp(\ddot{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}_t) \right) \ddot{\mathbf{x}}_i'$$

where $\phi(\cdot)$ is the standard normal density function, and $\mathbf{s}_{\text{ml,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}' = [\mathbf{s}_{\text{ml,om},1,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}', \mathbf{s}_{\text{ml,om},2,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}']$.

poisson

In the poisson case,

$$\mathbf{s}_{\text{ml,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t)\{y_i - \exp(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t')\}\mathbf{x}_i'$$

OM WNL

Here are the formulas for the $\mathbf{s}_{\text{nls,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}$ for each functional form choice.

linear

In the linear case,

$$\mathbf{s}_{\text{nls,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t)(y_i - \mathbf{x}_i\widehat{\boldsymbol{\beta}}_t)\mathbf{x}'_i$$

logit and probit

In the logit and probit cases,

$$\mathbf{s}_{\text{nls,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t) \left[g(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t) \left\{ y_i - G(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t) \right\} \right] \mathbf{x}_i$$

where $G(z)$ is the logistic cumulative distribution function for the logit, $G(z)$ is the normal cumulative distribution function for the probit, and $g(\cdot) = \{\partial G(z)\}/(\partial z)$ is the corresponding density function.

hetprobit

In the **hetprobit** case, there are two sets of sample score equations, $\mathbf{s}_{\text{nls,om},1,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}$ and $\mathbf{s}_{\text{nls,om},2,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}$:

$$\mathbf{s}_{\text{nls,om},1,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t) \left(\frac{\phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\}}{\exp(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t)} \left[y_i - \Phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\} \right] \right) \mathbf{x}'_i$$

and

$$\mathbf{s}_{\text{nls,om},2,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t) \left(\frac{\phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\}}{\exp(\mathbf{z}_i\widehat{\boldsymbol{\beta}}_t)} \mathbf{x}_i\widehat{\boldsymbol{\beta}}_t' \left[\Phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\} - y_i \right] \right) \mathbf{x}'_i$$

where $\phi(\cdot)$ is the standard normal density function, and $\mathbf{s}_{\text{nls,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}' = [\mathbf{s}_{\text{nls,om},1,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}', \mathbf{s}_{\text{nls,om},2,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}']$.

poisson

In the poisson case,

$$\mathbf{s}_{\text{nls,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t)\{y_i - \exp(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t)\} \exp(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t)\mathbf{x}'_i$$

Effect estimating functions

We now describe the sample estimating functions for the effect parameters, which vary over estimator and effect parameter.

RA estimators

RA estimators estimate the effect parameters using means of the observation-level predictions of the conditional means of the outcomes. There is no model for the conditional probability of treatment.

The RA estimators use unweighted QML estimators to estimate the parameters of the conditional mean model. In other words, the RA estimators use the sample estimating functions $s_{\text{ml,om},i}(\mathbf{x}_i, 1, \widehat{\beta})$, given above.

For the RA estimators, the vector of sample estimating functions is the concatenation of the sample estimating functions for the effect parameters with the sample estimating functions for the OM parameters. Algebraically,

$$s_{\text{ra},i}(\mathbf{x}_i, \widehat{\theta})' = s_{\text{ra},e,i}(\mathbf{x}_i, \widehat{\theta}, \widehat{\beta})', s_{\text{ml,om},i}(\mathbf{x}_i, 1, \widehat{\beta})'$$

The estimating functions $s_{\text{ra},e,i}(\mathbf{x}_i, \widehat{\theta}, \widehat{\beta})'$ vary over the effect parameter.

RA for POM

The RA estimators for the POM parameters estimate $\theta' = (\alpha', \beta')$ using two types of estimating equations: 1) those for the POM parameters α , and 2) those for the conditional-mean model parameters β_t in $\mu(\mathbf{x}, t, \beta_t)$.

The sample estimating functions for the $\widehat{\beta}_t$ are given in *OM estimating functions* above.

The elements of $s_{\text{ra},e,i}(\mathbf{x}_i, \widehat{\alpha}, \widehat{\beta})$ for the POM parameters α are given by

$$\mu(\mathbf{x}_i, t, \widehat{\beta}_t) - \widehat{\alpha}_t \tag{RAPOM}$$

RA for ATE

The RA estimators for the ATE parameters estimate $\theta' = (\tau', \beta')$ using two types of estimating equations: 1) those for the ATE parameters τ , and 2) those for the OM parameters β_t in $\mu(\mathbf{x}, t, \beta_t)$.

The sample estimating functions that determine the $\widehat{\beta}_t$ are given in *OM estimating functions* with $w_i(t) = 1$.

The elements of $s_{\text{ra},e,i}(\mathbf{x}_i, \widehat{\tau}, \widehat{\beta})$ for the ATE parameters τ are given by

$$\mu(\mathbf{x}_i, t, \widehat{\beta}_t) - \mu(\mathbf{x}_i, 0, \widehat{\beta}_t) - \widehat{\tau}_t \tag{RAATE}$$

RA for ATET

The RA estimators for the ATET parameters estimate $\theta' = (\delta', \beta')$ using two types of estimating equations: 1) those for the ATET parameters δ , and 2) those for the OM parameters β_t in $\mu(\mathbf{x}, t, \beta_t)$.

The sample estimating functions that determine the $\widehat{\beta}_t$ are given in *OM estimating functions* above with $w_i(t) = 1$.

The elements of $s_{\text{ra},e,i}(\mathbf{x}_i, \widehat{\delta}, \widehat{\beta})$ for the ATET parameters δ are given by

$$Nt_i(\tilde{t})/N_{\tilde{t}} \left\{ \mu(\mathbf{x}_i, t, \widehat{\beta}_t) - \mu(\mathbf{x}_i, 0, \widehat{\beta}_t) - \widehat{\delta}_t \right\} \tag{RAATET}$$

IPW estimators

IPW estimators estimate the effect parameters using means of the observed outcomes weighted by the inverse probability of treatment. There is no outcome model. The IPW estimators use QML estimators to estimate the parameters of the conditional probability model.

The vector of estimating functions is the concatenation of the estimating functions for the effect parameters with the estimating functions for the conditional-probability parameters. The sample estimating functions used by the IPW estimators are

$$s_{\text{ipw},i}(\mathbf{x}_i, \hat{\boldsymbol{\theta}})' = s_{\text{ipw},e,i}(\mathbf{x}_i, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\gamma}})', s_{\text{tm},i}(\mathbf{z}_i, 1, \hat{\boldsymbol{\gamma}})'$$

The estimating functions $s_{\text{ipw},e,i}(\mathbf{z}_i, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\gamma}})'$ vary over the effect parameter.

All the IPW estimators use normalized inverse-probability weights. These weights are not related to the weights $w_i(t)$ that were used in the OM equations. The functional form for the normalized inverse-probability weights varies over the effect parameters POM, ATE, and ATET.

The POM and ATE estimators use normalized inverse-probability weights. The unnormalized weights for individual i and treatment level t are $d_i(t) = t_i(t)/p(\mathbf{z}_i, t, \hat{\boldsymbol{\gamma}})$, and the normalized weights are $\bar{d}_i(t) = N_t d_i(t) / \sum_i^N d_i(t)$.

The ATET estimator uses normalized treatment-adjusted inverse-probability weights. The treatment-adjusted inverse-probability weights adjust the inverse-probability weights for the probability of getting the conditional treatment \tilde{t} . The unnormalized weights are $f_i = p(\mathbf{z}_i, \tilde{t}, \hat{\boldsymbol{\gamma}})/p(\mathbf{z}_i, t_i, \hat{\boldsymbol{\gamma}})$, and the normalized weights are $\bar{f}_i = N f_i / \sum_i^N f_i$.

The IPW effect estimators are weighted cell averages. The sample estimating functions used in POM estimators are the sample estimating functions from weighted OLS regression on treatment-cell indicators. The POM estimators use a full set of $q + 1$ of treatment indicator variables \mathbf{a}_i . (The i th observation on the k th variable in \mathbf{a}_i is 1 if i received treatment $k - 1$ and 0 otherwise, for $k \in \{1, 2, \dots, q + 1\}$.)

The sample estimating functions used in the ATE and the ATET estimators are the sample estimating functions from weighted OLS regression on treatment-cell indicators, excluding the indicator for the control level and including a constant term. The variables $\tilde{\mathbf{a}}_i$ used in the ATE and ATET sample estimating functions include q of treatment indicator variables and a variable that is always 1. (The first q variables in $\tilde{\mathbf{a}}_i$ are treatment indicators: the i th observation on the k th variable in $\tilde{\mathbf{a}}_i$ is 1 if i received treatment k and 0 otherwise, for $k \in \{1, 2, \dots, q\}$. The $(q + 1)$ th variable is always 1.) This definition of $\tilde{\mathbf{a}}_i$ sets the last treatment level to be the control; renaming the treatments handles the more general case allowed for by `teffects`.

Having defined \mathbf{a}_i and $\tilde{\mathbf{a}}_i$, we can give the sample estimating functions that the IPW estimators use for the effects parameters.

IPW for POM

We begin with the IPW estimators for the potential-outcome means. In this case, $\boldsymbol{\theta}' = (\boldsymbol{\alpha}', \boldsymbol{\gamma}')$.

The sample estimating functions for the $\hat{\boldsymbol{\gamma}}$ are given in [TM estimating functions](#) above.

The sample estimating functions for $\hat{\boldsymbol{\alpha}}$ are given by

$$s_{\text{ipw},e,i,t}(\mathbf{z}_i, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\gamma}})' = \bar{d}_i(t)(y_i - \mathbf{a}_i \hat{\boldsymbol{\alpha}}) \mathbf{a}_i' \quad (\text{IPWPOM})$$

IPW for ATE

The full parameter vector for the IPW estimators for the ATE is $\theta' = (\tau', \gamma')$.

The sample estimating functions for the $\hat{\gamma}$ are given in *TM estimating functions* above.

The sample estimating functions for $\hat{\tau}$ are given by

$$s_{\text{ipw},e,i,t}(\mathbf{z}_i, \hat{\tau}, \hat{\gamma})' = \bar{d}_i(t)(y_i - \tilde{\mathbf{a}}_i \hat{\tau}) \tilde{\mathbf{a}}_i' \quad (\text{IPWATE})$$

IPW for ATET

The full parameter vector for the IPW estimators for the ATET is $\theta' = (\delta', \gamma')$.

The sample estimating functions for the $\hat{\gamma}$ are given in *TM estimating functions* above.

The sample estimating functions for $\hat{\delta}$ are given by

$$s_{\text{ipw},e,i,t}(\mathbf{z}_i, \hat{\delta}, \hat{\gamma})' = \bar{f}_i(t)(y_i - \tilde{\mathbf{a}}_i \hat{\delta}) \tilde{\mathbf{a}}_i' \quad (\text{IPWATET})$$

AIPW estimators

This section documents the sample estimating functions used by the augmented inverse-probability-weighted (AIPW) estimators implemented in **teffects**. These AIPW estimators are efficient-influence-function estimators as discussed in [TE] **teffects intro** and [TE] **teffects intro advanced**. The **teffects** implementation was primarily inspired by Cattaneo, Drukker, and Holland (2013), which was based on Cattaneo (2010). Tan (2010) was influential by identifying the implemented weighted nonlinear least-squares estimator as having relatively good properties when both the conditional mean function and the conditional probability function are misspecified.

The AIPW estimating functions for the treatment parameters include terms from a conditional probability model and from a conditional mean model for the outcome.

The sample-estimation-equations vector has three parts for the AIPW estimators:

$$s_{\text{aipw},i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta})' = [s_{\text{aipw},e,i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta})', s_{\text{aipw},\text{tm},i}(\mathbf{z}_i, \hat{\gamma})', s_{\text{aipw},\text{om},i}\{\mathbf{x}_i, w_i(t), \hat{\beta}\}]'$$

The sample estimating functions for the parameters of the TM are the $s_{\text{tm},i}(\mathbf{z}_i, \hat{\gamma})$ given in *TM estimating functions* above.

teffects aipw implements three AIPW estimators: the standard AIPW estimator, the NLS AIPW estimator, and the WNLS AIPW estimator.

The three AIPW estimators differ in how they estimate the parameters of the OM.

By default, **teffects aipw** uses the standard AIPW estimator that estimates the parameters of the OM by the unweighted ML estimator, whose sample estimating functions, $s_{\text{ml},\text{om},i}(\mathbf{x}_i, 1, \hat{\beta})$, are given in *OM estimating functions* above.

When the **nls** option is specified, **teffects aipw** uses the NLS AIPW estimator that estimates the parameters of the OM by the unweighted NLS estimator, whose sample estimating functions, $s_{\text{nls},\text{om},i}(\mathbf{x}_i, 1, \hat{\beta})$, are given in *OM estimating functions* above.

When the `wnls` option is specified, `teffects aipw` uses the WNLS AIPW estimator that estimates the parameters of the OM by the WNLS estimator, whose sample estimating functions, $\mathbf{s}_{\text{wnls,om},i}\{\mathbf{x}_i, \tilde{w}_i(t), \hat{\boldsymbol{\beta}}\}$, are given in *OM estimating functions* above. The weights for person i in treatment level t are

$$\tilde{w}_i(t) = \frac{t_i(t)}{p(\mathbf{z}_i, t, \hat{\gamma})} \left\{ \frac{t_i(t)}{p(\mathbf{z}_i, t, \hat{\gamma})} - 1 \right\} \quad (\text{WNLSW})$$

Now we discuss the sample estimating functions for the effect parameters, the $\mathbf{s}_{e,i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\boldsymbol{\theta}})$.

AIPW for POM

We begin with the AIPW estimators for the potential-outcome means. In this case, $\boldsymbol{\theta}' = (\boldsymbol{\alpha}', \boldsymbol{\gamma}', \boldsymbol{\beta}')$, and the elements of $\mathbf{s}_{\text{aipw},e,i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\boldsymbol{\theta}})$ are given by

$$\frac{t_i(t)}{p(\mathbf{z}_i, t, \hat{\gamma})} y_i - \mu(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_t) \left\{ \frac{t_i(t)}{p(\mathbf{z}_i, t, \hat{\gamma})} - 1 \right\} - \alpha_t$$

AIPW for ATE

The AIPW estimators for the ATE estimate $\boldsymbol{\theta}' = (\boldsymbol{\tau}', \boldsymbol{\gamma}', \boldsymbol{\beta}')$, and the elements of $\mathbf{s}_{\text{aipw},e,i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\boldsymbol{\theta}})$ are given by

$$\begin{aligned} & \frac{t_i(t)}{p(\mathbf{z}_i, t, \hat{\gamma})} y_i - \mu(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_t) \left\{ \frac{t_i(t)}{p(\mathbf{z}_i, t, \hat{\gamma})} - 1 \right\} - \tau_0 \text{ if } t = 0 \\ & \frac{t_i(t)}{p(\mathbf{z}_i, t, \hat{\gamma})} y_i - \mu(\mathbf{x}_i, \hat{\boldsymbol{\beta}}_t) \left\{ \frac{t_i(t)}{p(\mathbf{z}_i, t, \hat{\gamma})} - 1 \right\} - \tau_t - \tau_0 \text{ if } t > 0 \end{aligned}$$

IPWRA estimators

The IPWRA estimators combine inverse-probability weighting (IPW) with regression-adjustment (RA) methods. The sample estimating functions for IPWRA include sample estimating functions from both RA and IPW. The vector of sample estimating functions is

$$\mathbf{s}_{\text{ipwra},i}(\mathbf{x}_i, \hat{\boldsymbol{\theta}})' = \mathbf{s}_{\text{ra},e,i}(\mathbf{x}_i, \hat{\boldsymbol{\vartheta}}, \hat{\boldsymbol{\beta}})' , \mathbf{s}_{\text{ml,om},i}\{\mathbf{x}_i, w_i(j), \hat{\boldsymbol{\beta}}\}' , \mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \hat{\gamma})'$$

where $\hat{\boldsymbol{\theta}}' = (\hat{\boldsymbol{\vartheta}}', \hat{\boldsymbol{\beta}}', \hat{\boldsymbol{\gamma}}')$, $\hat{\boldsymbol{\vartheta}} = \hat{\boldsymbol{\alpha}}$ for POM, $\hat{\boldsymbol{\vartheta}} = \hat{\boldsymbol{\tau}}_t$ for ATE, and $\hat{\boldsymbol{\vartheta}} = \hat{\boldsymbol{\delta}}_t$ for ATET. The sample estimating functions, $\mathbf{s}_{\text{ra},e,i}(\mathbf{x}_i, \hat{\boldsymbol{\vartheta}}, \hat{\boldsymbol{\beta}})$, for POM, ATE, and ATET are given in equations (RAPOM), (RAATE), and (RAATET). For the OM sample estimating functions, $\mathbf{s}_{\text{ml,om},i}(\cdot)$, we replace the RA unity weights, $w_i(t) = 1$, with $d_i(j)$ for POM or ATE and \bar{f}_i for ATET, the normalized inverse-probability weights described in *IPW estimators* above. The specific form of the OM sample estimating functions are given in *OM estimating functions* above. The TM sample estimating functions are given in *TM estimating functions* above.

References

- 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.
- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.
- Newey, W. K., and D. McFadden. 1994. Large sample estimation and hypothesis testing. In Vol. 4 of *Handbook of Econometrics*, ed. R. F. Engle and D. L. McFadden, 2111–2245. Amsterdam: Elsevier.
- Stefanski, L. A., and D. D. Boos. 2002. The calculus of M-estimation. *American Statistician* 56: 29–38.
- Tan, Z. 2010. Bounded, efficient and doubly robust estimation with inverse weighting. *Biometrika* 97: 661–682.
- Tsiatis, A. A. 2006. *Semiparametric Theory and Missing Data*. New York: Springer.
- van der Vaart, A. W. 1998. *Asymptotic Statistics*. Cambridge: Cambridge University Press.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

- [TE] **teffects postestimation** — Postestimation tools for teffects
- [TE] **teffects** — Treatment-effects estimation for observational data
- [TE] **teffects ipwra** — Inverse-probability-weighted regression adjustment
- [R] **heckman** — Heckman selection model
- [R] **hetprobit** — Heteroskedastic probit model
- [R] **logit** — Logistic regression, reporting coefficients
- [R] **mlogit** — Multinomial (polytomous) logistic regression
- [R] **poisson** — Poisson regression
- [R] **probit** — Probit regression
- [R] **regress** — Linear regression
- [U] **20 Estimation and postestimation commands**

Title

teffects ipw — Inverse-probability weighting

Syntax	Menu	Description	Options
Remarks and examples	Stored results	Methods and formulas	Reference
Also see			

Syntax

```
teffects ipw (ovar) (tvar tmvarlist [, tmodel noconstant]) [if] [in] [weight]  
[ , stat options]
```

<i>tmodel</i>	Description
---------------	-------------

Model

<code>logit</code>	logistic treatment model; the default
<code>probit</code>	probit treatment model
<code>hetprobit(<i>varlist</i>)</code>	heteroskedastic probit treatment model

tmodel specifies the model for the treatment variable.

For multivariate treatments, only `logit` is available and multinomial logit is used.

<i>stat</i>	Description
-------------	-------------

Stat

<code>ate</code>	estimate average treatment effect in population; the default
<code>atet</code>	estimate average treatment effect on the treated
<code>pomeans</code>	estimate potential-outcome means

<i>options</i>	Description
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>robust</code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>aequations</code>	display auxiliary-equation results
<code>display_options</code>	control column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
Advanced	
<code>pstolerance(#)</code>	set tolerance for overlap assumption
<code>osample(<i>newvar</i>)</code>	<i>newvar</i> identifies observations that violate the overlap assumption
<code>control(# <i>label</i>)</code>	specify the level of <i>tvar</i> that is the control
<code>tlevel(# <i>label</i>)</code>	specify the level of <i>tvar</i> that is the treatment
<code>coeflegend</code>	display legend instead of statistics

tvar must contain integer values representing the treatment levels.

tmvarlist may contain factor variables; see [U] 11.4.3 **Factor variables**.

`bootstrap`, `by`, `jackknife`, and `statsby` are allowed; see [U] 11.1.10 **Prefix commands**.

`fweights` and `iweights` are allowed; see [U] 11.1.6 **weight**.

`coeflegend` does not appear in the dialog box.

See [U] 20 **Estimation and postestimation commands** for more capabilities of estimation commands.

Menu

Statistics > Treatment effects > Inverse-probability weights (IPW)

Description

`teffects ipw` estimates treatment effects from observational data via inverse-probability weighting (IPW). IPW estimators use estimated probability weights to correct for the missing-data problem arising from the fact that each subject is observed in only one of the potential outcomes. `teffects ipw` offers several choices for the functional form of the model used to predict treatment.

See [TE] **teffects intro** or [TE] **teffects intro advanced** for more information about these estimators.

Options

Model

`noconstant`; see [R] **estimation options**.

Stat

stat is one of three statistics: `ate`, `atet`, or `pomeans`. `ate` is the default.

`ate` specifies that the average treatment effect be estimated.

`atet` specifies that the average treatment effect on the treated be estimated.

`pomeans` specifies that the potential-outcome means for each treatment level be estimated.

SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`) and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce_option](#).

Reporting

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

`aequations` specifies that the results for the outcome-model or the treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

`display_options`: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `novlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `no1stretch`; see [R] [estimation options](#).

Maximization

`maximize_options`: `iterate(#)`, `[no]log`, and `from(init_specs)`; see [R] [maximize](#). These options are seldom used.

`init_specs` is one of

`matname [, skip copy]`

`# [, # ...] , copy`

Advanced

`pstolerance(#)` specifies the tolerance used to check the overlap assumption. The default value is `pstolerance(1e-5)`. `teffects` will exit with an error if an observation has an estimated propensity score smaller than that specified by `pstolerance()`.

`osample(newvar)` specifies that indicator variable `newvar` be created to identify observations that violate the overlap assumption.

`control(# | label)` specifies the level of `tvar` that is the control. The default is the first treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with statistic `pomeans`. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(# | label)` specifies the level of `tvar` that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

The following option is available with `teffects ipw` but is not shown in the dialog box:

`coeflegend`; see [R] [estimation options](#).

Remarks and examples

Remarks are presented under the following headings:

[Overview](#)

[Video example](#)

Overview

IPW estimators use estimated probability weights to correct for the missing-data problem arising from the fact that each subject is observed in only one of the potential outcomes. IPW estimators use a two-step approach to estimating treatment effects:

1. They estimate the parameters of the treatment model and compute the estimated inverse-probability weights.
2. They use the estimated inverse-probability weights to compute weighted averages of the outcomes for each treatment level. The contrasts of these weighted averages provide the estimates of the ATEs. Using this weighting scheme corrects for the missing potential outcomes.

These steps produce consistent estimates of the effect parameters because the treatment is assumed to be independent of the potential outcomes after conditioning on the covariates. The overlap assumption ensures that predicted inverse-probability weights do not get too large. In fact, `teffects ipw` uses an estimation technique that implements both steps at once so that we do not need to correct the standard errors in the second step to reflect the uncertainty associated with the predicted treatment probabilities.

We will illustrate the use of `teffects ipw` by using data from a study of the effect of a mother's smoking status during pregnancy (`mbsmoke`) on infant birthweight (`bweight`) as reported by Cattaneo (2010). This dataset also contains information about each mother's age (`mage`), education level (`medu`), marital status (`mmarried`), whether the first prenatal exam occurred in the first trimester (`prenatal1`), and whether this baby was the mother's first birth (`fbaby`).

▷ Example 1: Estimating the ATE

We begin by using `teffects ipw` to estimate the average treatment effect of smoking on birthweight. We will use a probit model to predict treatment status, using `prenatal1`, `mmarried`, `mage`, the square of `mage`, and `fbaby` as explanatory variables:

```
. use http://www.stata-press.com/data/r13/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ipw (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu, probit)
Iteration 0:  EE criterion = 4.622e-21
Iteration 1:  EE criterion = 8.070e-26
Treatment-effects estimation      Number of obs      =      4642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: probit
```

bweight	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
ATE						
mbsmoke (smoker vs nonsmoker)	-230.6886	25.81524	-8.94	0.000	-281.2856	-180.0917
POmean						
mbsmoke nonsmoker	3403.463	9.571369	355.59	0.000	3384.703	3422.222

Smoking causes infants' birthweights to be reduced by an average of 231 grams from the average of 3,403 grams for infants of mothers who do not smoke.

◀

Sometimes, we are mainly concerned about those subjects that did in fact receive treatment, and we want to know how much the outcome changes as a result of treatment for that subpopulation. The ATET provides us with the answer. Moreover, the ATET can be estimated using weaker assumptions than are required to estimate the ATE; see [TE] [teffects intro advanced](#).

▶ Example 2: Estimating the ATET

```
. teffects ipw (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu, probit),
> atet
Iteration 0:  EE criterion = 4.634e-21
Iteration 1:  EE criterion = 6.657e-27
Treatment-effects estimation      Number of obs      =      4642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: probit
```

bweight	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
ATET						
mbsmoke (smoker vs nonsmoker)	-225.1773	23.66458	-9.52	0.000	-271.559	-178.7955
POmean						
mbsmoke nonsmoker	3362.837	14.20149	236.79	0.000	3335.003	3390.671

Among smoking mothers, smoking causes infants' birthweights to be reduced by an average of 225 grams from the average of 3,363 grams that would have occurred if these mothers had not smoked.

◀

We often express statistics as percentages to alleviate scaling issues and aid interpretation. In the present context, we may wish to express an ATE as a percentage of the untreated POM to gain a more intuitive measure of the effect of treatment.

▶ Example 3: Reporting the ATE as a percentage

Here we use the same model as in example 1, but we report the ATE as a percentage of the mean birthweight that would occur if no mothers smoke. First, we use `teffects ipw` to fit the model. We use the `coeflegend` option so that `teffects ipw` reports the names of the parameters. Then we use `nlcom` to obtain the statistic we want along with its delta-method-based standard error. We type

```
. teffects ipw (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu, probit),
> coeflegend
Iteration 0:  EE criterion = 4.622e-21
Iteration 1:  EE criterion = 8.070e-26
Treatment-effects estimation      Number of obs      =      4642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: probit
```

bweight	Coef.	Legend
ATE		
mbsmoke (smoker vs nonsmoker)	-230.6886	_b[ATE:r1vs0.mbsmoke]
P0mean		
mbsmoke nonsmoker	3403.463	_b[P0mean:r0.mbsmoke]

```
. nlcom _b[ATE:r1vs0.mbsmoke] / _b[P0mean:r0.mbsmoke]
      _nl_1:  _b[ATE:r1vs0.mbsmoke] / _b[P0mean:r0.mbsmoke]
```

bweight	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
_nl_1	-.0677806	.0075169	-9.02	0.000	-.0825133 -.0530478

Smoking reduces the average birthweight of an infant by a statistically significant 6.8%.

◀

Video example

[Treatment effects: Inverse probability weights](#)

Stored results

`teffects ipw` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level j
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	<code>teffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(subcmd)</code>	<code>ipw</code>
<code>e(tmodel)</code>	<code>logit</code> , <code>probit</code> , or <code>hetprobit</code>
<code>e(stat)</code>	statistic estimated, <code>ate</code> , <code>atet</code> , or <code>pomeans</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. Err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

Methods and formulas

`teffects ipw` implements a smooth treatment-effects estimator. All smooth treatment-effects estimators are documented in *Methods and formulas* of [TE] [teffects aipw](#).

Reference

Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154.

Also see

- [TE] **teffects postestimation** — Postestimation tools for teffects
- [TE] **teffects** — Treatment-effects estimation for observational data
- [R] **hetprobit** — Heteroskedastic probit model
- [R] **logit** — Logistic regression, reporting coefficients
- [R] **mlogit** — Multinomial (polytomous) logistic regression
- [R] **probit** — Probit regression
- [U] **20 Estimation and postestimation commands**

Title

teffects ipwra — Inverse-probability-weighted regression adjustment

Syntax	Menu	Description	Options
Remarks and examples	Stored results	Methods and formulas	Reference
Also see			

Syntax

```
teffects ipwra (ovar omvarlist [, omodel noconstant])  
  (tvar tmvarlist [, tmodel noconstant]) [if] [in] [weight]  
  [, stat options]
```

<i>omodel</i>	Description
---------------	-------------

Model

<code>linear</code>	linear outcome model; the default
<code>logit</code>	logistic outcome model
<code>probit</code>	probit outcome model
<code>hetprobit(<i>varlist</i>)</code>	heteroskedastic probit outcome model
<code>poisson</code>	exponential outcome model

omodel specifies the model for the outcome variable.

<i>tmodel</i>	Description
---------------	-------------

Model

<code>logit</code>	logistic treatment model; the default
<code>probit</code>	probit treatment model
<code>hetprobit(<i>varlist</i>)</code>	heteroskedastic probit treatment model

tmodel specifies the model for the treatment variable.

For multivariate treatments, only `logit` is available and multinomial `logit` is used.

<i>stat</i>	Description
-------------	-------------

Stat

<code>ate</code>	estimate average treatment effect in population; the default
<code>atet</code>	estimate average treatment effect on the treated
<code>pomeans</code>	estimate potential-outcome means

<i>options</i>	Description
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>robust</code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>aequations</code>	display auxiliary-equation results
<code>display_options</code>	control column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
Advanced	
<code>pstolerance(#)</code>	set tolerance for overlap assumption
<code>osample(<i>newvar</i>)</code>	<i>newvar</i> identifies observations that violate the overlap assumption
<code>control(# <i>label</i>)</code>	specify the level of <i>tvar</i> that is the control
<code>tlevel(# <i>label</i>)</code>	specify the level of <i>tvar</i> that is the treatment
<code>coeflegend</code>	display legend instead of statistics

tvar must contain integer values representing the treatment levels.

omvarlist and *tmvarlist* may contain factor variables; see [U] 11.4.3 Factor variables.

`bootstrap`, `by`, `jackknife`, and `statsby` are allowed; see [U] 11.1.10 Prefix commands.

`fweights` and `iweights` are allowed; see [U] 11.1.6 weight.

`coeflegend` does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Menu

Statistics > Treatment effects > Doubly robust estimators > Regress adjustment and IPW

Description

`teffects ipwra` estimates treatment effects from observational data by using inverse-probability-weighted regression-adjustment (IPWRA) estimators. IPWRA estimators use weighted regression coefficients to compute averages of treatment-level predicted outcomes, where the weights are the estimated inverse probabilities of treatment. The contrasts of these averages provide the estimated treatment effects. IPWRA estimators have the double-robust property. `teffects ipwra` offers several choices for the functional forms of the outcome model and of the treatment model.

See [TE] [teffects intro](#) or [TE] [teffects intro advanced](#) for more information about these estimators.

Options

Model

`noconstant`; see [R] [estimation options](#).

Stat

stat is one of three statistics: `ate`, `atet`, or `pomeans`. `ate` is the default.

`ate` specifies that the average treatment effect be estimated.

`atet` specifies that the average treatment effect on the treated be estimated.

`pomeans` specifies that the potential-outcome means for each treatment level be estimated.

SE/Robust

`vce(vctype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`) and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce_option](#).

Reporting

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

`aequations` specifies that the results for the outcome-model or the treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

`display_options`: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `no!stretch`; see [R] [estimation options](#).

Maximization

`maximize_options`: `iterate(#)`, `[no]log`, and `from(init_specs)`; see [R] [maximize](#). These options are seldom used.

`init_specs` is one of

`matname` [, skip copy]

`#` [, # ...] , copy

Advanced

`pstolerance(#)` specifies the tolerance used to check the overlap assumption. The default value is `pstolerance(1e-5)`. `teffects` will exit with an error if an observation has an estimated propensity score smaller than that specified by `pstolerance()`.

`osample(newvar)` specifies that indicator variable `newvar` be created to identify observations that violate the overlap assumption.

`control(# | label)` specifies the level of `tvar` that is the control. The default is the first treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with statistic `pomeans`. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(# | label)` specifies the level of `tvar` that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

The following option is available with `teffects ipwra` but is not shown in the dialog box:

`coeflegend`; see [R] [estimation options](#).

Remarks and examples

Remarks are presented under the following headings:

[Overview](#)

[Video example](#)

Overview

IPWRA estimators use probability weights to obtain outcome-regression parameters that account for the missing-data problem arising from the fact that each subject is observed in only one of the potential outcomes. The adjusted outcome-regression parameters are used to compute averages of treatment-level predicted outcomes. The contrasts of these averages provide estimates of the treatment effects.

IPWRA estimators use a model to predict treatment status, and they use another model to predict outcomes. Because IPWRA estimators have the double-robust property, only one of the two models must be correctly specified for the IPWRA estimator to be consistent.

IPWRA estimators use a three-step approach to estimating treatment effects:

1. They estimate the parameters of the treatment model and compute inverse-probability weights.
2. Using the estimated inverse-probability weights, they fit weighted regression models of the outcome for each treatment level and obtain the treatment-specific predicted outcomes for each subject.
3. They compute the means of the treatment-specific predicted outcomes. The contrasts of these averages provide the estimates of the ATEs. By restricting the computations of the means to the subset of treated subjects, we can obtain the ATETs.

These steps produce consistent estimates of the effect parameters because the treatment is assumed to be independent of the potential outcomes after conditioning on the covariates. The overlap assumption ensures that predicted inverse-probability weights do not get too large. The standard errors reported by `teffects ipwra` correct for the three-step process. See [\[TE\] teffects intro](#) or [\[TE\] teffects intro advanced](#) for more information about this estimator.

We will illustrate the use of `teffects ipwra` by using data from a study of the effect of a mother's smoking status during pregnancy (`mbsmoke`) on infant birthweight (`bweight`) as reported by [Cattaneo \(2010\)](#). This dataset also contains information about each mother's age (`mage`), education level (`medu`), marital status (`mmarried`), whether the first prenatal exam occurred in the first trimester (`prenatal1`), and whether this baby was the mother's first birth (`fbaby`).

► Example 1: Estimating the ATE

We begin by using `teffects ipwra` to estimate the average treatment effect of smoking on birthweight. We will use a probit model to predict treatment status as a function of `mmarried`, `mage`, and `fbaby`; to maximize the predictive power of this model, we use factor-variable notation to incorporate quadratic effects of the mother's age, the only continuous covariate in our model. We will use linear regression (the default) to model birthweight, using `prenatal1`, `mmarried`, `mage`, and `fbaby` as explanatory variables. We type

```

. use http://www.stata-press.com/data/r13/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ipwra (bweight prenatal1 mmarried mage fbaby)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, probit)

Iteration 0:  EE criterion = 9.482e-21
Iteration 1:  EE criterion = 1.200e-25

Treatment-effects estimation          Number of obs      =      4642
Estimator      : IPW regression adjustment
Outcome model  : linear
Treatment model: probit

```

	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
ATE						
mbsmoke (smoker vs nonsmoker)	-229.9671	26.62668	-8.64	0.000	-282.1544	-177.7798
POmean						
mbsmoke nonsmoker	3403.336	9.57126	355.58	0.000	3384.576	3422.095

Smoking causes infants' birthweights to be reduced by an average of 230 grams from the average of 3,403 grams for infants of mothers who do not smoke.

◀

By default, `teffects ipwra` displays the ATE and untreated POM. We can specify the `pomeans` option to display both the treated and untreated POMs, and we can use the `aequations` option to display the regression model coefficients used to predict the POMs as well as the coefficients from the model used to predict treatment.

▷ Example 2: Displaying the POMs and equations

```

. use http://www.stata-press.com/data/r13/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ipwra (bweight prenatal1 mmarried mage fbaby)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, probit), pomeans aequations

Iteration 0:  EE criterion = 9.482e-21
Iteration 1:  EE criterion = 1.157e-25

Treatment-effects estimation      Number of obs      =      4642
Estimator      : IPW regression adjustment
Outcome model  : linear
Treatment model: probit

```

bweight	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
POmeans						
mbsmoke						
nonsmoker	3403.336	9.57126	355.58	0.000	3384.576	3422.095
smoker	3173.369	24.86997	127.60	0.000	3124.624	3222.113
OME0						
prenatal1	67.98549	28.78428	2.36	0.018	11.56933	124.4017
mmarried	155.5893	26.46903	5.88	0.000	103.711	207.4677
mage	2.893051	2.134788	1.36	0.175	-1.291056	7.077158
fbaby	-71.9215	20.39317	-3.53	0.000	-111.8914	-31.95162
_cons	3194.808	55.04911	58.04	0.000	3086.913	3302.702
OME1						
prenatal1	34.76923	43.18534	0.81	0.421	-49.87248	119.4109
mmarried	124.0941	40.29775	3.08	0.002	45.11193	203.0762
mage	-5.068833	5.954425	-0.85	0.395	-16.73929	6.601626
fbaby	39.89692	56.82072	0.70	0.483	-71.46966	151.2635
_cons	3175.551	153.8312	20.64	0.000	2874.047	3477.054
TME1						
mmarried	-.6484821	.0554173	-11.70	0.000	-.757098	-.5398663
mage	.1744327	.0363718	4.80	0.000	.1031452	.2457202
c.mage#						
c.mage	-.0032559	.0006678	-4.88	0.000	-.0045647	-.0019471
fbaby	-.2175962	.0495604	-4.39	0.000	-.3147328	-.1204595
medu	-.0863631	.0100148	-8.62	0.000	-.1059917	-.0667345
_cons	-1.558255	.4639691	-3.36	0.001	-2.467618	-.6488926

◀

As is well known, the standard probit model assumes that the error terms in the latent-utility framework are homoskedastic; the model is not robust to departures from this assumption. An alternative is to use the heteroskedastic probit model, which explicitly models the error variance as a function of a set of variables.

▷ Example 3: Heteroskedastic probit treatment model

Here we use the variables as before, but we use a heteroskedastic probit model to predict treatment status, modeling the heteroskedasticity as a quadratic function of the mother's age:

```

. teffects ipwra (bweight prenatal1 mmarried fbaby c.mage)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, hetprobit(c.mage##c.mage)),
> aequations

Iteration 0:   EE criterion = 1.312e-10
Iteration 1:   EE criterion = 1.526e-20

Treatment-effects estimation           Number of obs   =       4642
Estimator      : IPW regression adjustment
Outcome model  : linear
Treatment model: heteroskedastic probit

```

	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
ATE						
mbsmoke (smoker vs nonsmoker)	-229.6322	26.33453	-8.72	0.000	-281.2469	-178.0174
POMean						
mbsmoke nonsmoker	3403.74	9.545798	356.57	0.000	3385.03	3422.449
OME0						
prenatal1	64.95125	28.62162	2.27	0.023	8.853899	121.0486
mmarried	154.2297	26.45868	5.83	0.000	102.3717	206.0878
fbaby	-71.61131	20.33774	-3.52	0.000	-111.4725	-31.75006
mage	3.010148	2.133812	1.41	0.158	-1.172047	7.192344
_cons	3195.355	55.05451	58.04	0.000	3087.45	3303.26
OME1						
prenatal1	38.55274	43.57026	0.88	0.376	-46.8434	123.9489
mmarried	126.3377	40.7398	3.10	0.002	46.48922	206.1863
fbaby	45.43547	56.44832	0.80	0.421	-65.20121	156.0721
mage	-6.069913	5.952513	-1.02	0.308	-17.73662	5.596798
_cons	3195.795	152.3979	20.97	0.000	2897.101	3494.49
TME1						
mmarried	-.0295523	.0238304	-1.24	0.215	-.0762589	.0171544
mage	.0157893	.0105203	1.50	0.133	-.0048302	.0364088
c.mage#						
c.mage	-.0002837	.0001896	-1.50	0.135	-.0006553	.0000878
fbaby	-.0093306	.0079804	-1.17	0.242	-.024972	.0063107
medu	-.0036773	.0030235	-1.22	0.224	-.0096033	.0022487
_cons	-.1822201	.1177454	-1.55	0.122	-.4129969	.0485567
TME1_lnsigma						
mage	-.2211492	.0629973	-3.51	0.000	-.3446217	-.0976767
c.mage#						
c.mage	.0037613	.0012413	3.03	0.002	.0013285	.0061942

The estimated ATE and base-level POM are essentially the same as those produced by the model that used a homoskedastic probit.

Video example

Treatment effects: Inverse probability weights with regression adjustment

Stored results

`teffects ipwra` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level j
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	<code>teffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(subcmd)</code>	<code>ipwra</code>
<code>e(tmodel)</code>	<code>logit</code> , <code>probit</code> , or <code>hetprobit</code>
<code>e(omodel)</code>	<code>linear</code> , <code>logit</code> , <code>probit</code> , <code>hetprobit</code> , or <code>poisson</code>
<code>e(stat)</code>	statistic estimated, <code>ate</code> , <code>atet</code> , or <code>pomeans</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<code>vcetype</code> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. Err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

Methods and formulas

`teffects ipwra` implements a smooth treatment-effects estimator. All smooth treatment-effects estimators are documented in *Methods and formulas* of [TE] **teffects aipw**.

Reference

Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154.

Also see

- [TE] **teffects postestimation** — Postestimation tools for teffects
- [TE] **teffects** — Treatment-effects estimation for observational data
- [TE] **teffects aipw** — Augmented inverse-probability weighting
- [R] **heckman** — Heckman selection model
- [R] **hetprobit** — Heteroskedastic probit model
- [R] **logit** — Logistic regression, reporting coefficients
- [R] **mlogit** — Multinomial (polytomous) logistic regression
- [R] **poisson** — Poisson regression
- [R] **probit** — Probit regression
- [R] **regress** — Linear regression
- [U] **20 Estimation and postestimation commands**

Title

teffects multivalued — Multivalued treatment effects

[Description](#)

[Remarks and examples](#)

[References](#)

[Also see](#)

Description

This entry discusses the use of `teffects` when the treatment is multivalued. This entry presumes you are already familiar with the potential-outcome framework and the use of the `teffects` commands with binary treatments. See [\[TE\] `teffects intro`](#) or [\[TE\] `teffects intro advanced`](#) for more information.

Remarks and examples

Remarks are presented under the following headings:

[Introduction](#)

[Parameters and notation](#)

[Illustrating multivalued treatments](#)

[Examples](#)

Introduction

When the treatment is binary, each subject could either receive the treatment or not receive the treatment. In contrast, multivalued treatments refer to cases in which each subject could receive one of several different treatments or else not receive treatment at all. For example, in testing the efficacy of a drug, a patient could receive a 10 milligram (mg) dose, a 20 mg dose, a 30 mg dose, or no dose at all. We first want to be able to compare a patient receiving the 10 mg dose with a patient receiving no dose, a patient receiving the 20 mg dose with a patient receiving no dose, and a patient receiving the 30 mg dose with a patient receiving no dose. Once we can make those comparisons, we can then, for example, compare the efficacy of a 30 mg dose with that of a 20 mg dose or a 10 mg dose.

To highlight an example in economics, we consider an unemployed person who could participate in a comprehensive skills training program, attend a one-day workshop that helps job seekers write their resumés, or choose not to participate in either. We want to know how effective each of those programs is relative to not participating; once we know that, we can then compare the effectiveness of the comprehensive program with that of the one-day program.

Multivalued treatments increase the number of parameters that must be estimated and complicate the notation. Fortunately, however, using the `teffects` commands is not much more difficult with multivalued treatments than with binary treatments.

You can use `teffects ra`, `teffects ipw`, `teffects ipwra`, and `teffects aipw` to estimate multivalued treatment effects. However, the theory developed in [Abadie and Imbens \(2006, 2012\)](#) has not been extended to handle multivalued treatments, so you cannot use `teffects nmatch` or `teffects psmatch` in these cases.

[Cattaneo \(2010\)](#), [Imbens \(2000\)](#), and [Wooldridge \(2010, sec. 21.6.3\)](#) discuss aspects of treatment-effect estimation with multivalued treatments.

Parameters and notation

We denote the potential outcome that subject i would obtain if given treatment-level t as y_{ti} , where y_{ti} is the realization of the random variable y_t . Throughout this entry, i subscripts denote realizations of the corresponding unsubscripted random variables. We again let y_0 denote the potential outcome of a subject who did not receive any treatment. To handle the case of multivalued treatments, we extend the definition of the unobservable, individual-level treatment effects to be $y_t - y_0$ for $t \in \{1, \dots, q\}$.

As in the binary-valued case, we again focus on three parameters of interest: the average treatment effect (ATE), the potential-outcome mean (POM), and the average treatment effect on the treated (ATET).

ATE The ATE is the average effect of giving each individual treatment t instead of treatment 0:

$$\text{ATE}_t = E(y_t - y_0)$$

POM The POM for each treatment level is an average of each potential outcome:

$$\text{POM}_t = E(y_t)$$

ATET The ATET is the average effect among those subjects that receive treatment level \tilde{t} of giving each subject treatment \tilde{t} instead of treatment 0:

$$\text{ATET}_{\tilde{t}, \tilde{t}} = E\{(y_{\tilde{t}} - y_0) | t = \tilde{t}\}$$

The extra notation required to define the ATET in this case indicates the difficulties surrounding this parameter.

Defining the ATET in the multivalued treatment case requires three different treatment levels: \tilde{t} defines the treatment level of the treated potential outcome; 0 is the treatment level of the control potential outcome; and $t = \tilde{t}$ restricts the expectation to include only those individuals who actually receive treatment level \tilde{t} .

Illustrating multivalued treatments

To illustrate the concept of a potential outcome and the parameters we would like to estimate, we consider the following table:

y	t	y_0	y_1	y_2
-0.50	0	-0.50	1.06	1.93
2.42	1	2.13	2.42	2.43
3.15	2	1.26	2.57	3.15
-0.39	0	-0.39	-0.18	0.52
2.22	2	-0.24	-0.01	2.22

We observe the outcome y as well as the treatment indicator t . There are three levels of treatment: 0, 1, or 2. Ideally, we would observe y_0 , y_1 , and y_2 , but in fact all we have is y . In the first row, the subject received treatment level 0, so $y = y_0$ for that subject. In the last row, the subject received treatment 2, so $y = y_2$. We reiterate that we do not actually observe y_0 , y_1 , or y_2 .

If we did have data on y_0 , y_1 , and y_2 , then we could define subject-level treatment variables $te_1 = y_1 - y_0$ and $te_2 = y_2 - y_0$. Here we would be following the convention of taking treatment level 0 to be the control level. The following table adds these two variables:

y	t	y_0	y_1	y_2	te_1	te_2
-0.50	0	-0.50	1.06	1.93	1.56	2.43
2.42	1	2.13	2.42	2.43	0.29	0.30
3.15	2	1.26	2.57	3.15	1.31	1.89
-0.39	0	-0.39	-0.18	0.52	0.21	0.91
2.22	2	-0.24	-0.01	2.22	0.23	2.46

Once we have te_1 and te_2 , obtaining the ATEs is straightforward. The ATE of going from treatment 0 to treatment 1 is simply the mean of the five entries in the column labeled te_1 , which here works out to 0.72. Going from treatment level 0 to treatment level 1 causes the outcome to increase an average of 0.72. Similarly, the ATE of going from treatment 0 to treatment 2 is the mean of the entries in the column labeled te_2 , which is 1.60. Exposing all subjects to treatment level 2 would cause the outcome to rise by an average of 1.60 relative to the outcome obtained by exposing them to treatment level 0.

The ATET is the average difference in the potential outcomes among those that get a particular treatment level. To compute this, we must specify two treatment levels: the actual treatment level the subjects we are interested in received as well as the treatment level we want to compare them with. For example, suppose we are interested in the ATET of going from treatment 0 to treatment 1 for those who received treatment 0. This ATET is the average of te_1 for those subjects for which $t = 0$. Here that ATET is just $(1.56 + 0.21)/2 \approx 0.89$. If we exposed the subjects who received treatment 0 to treatment 1 instead, the outcome would increase an average of 0.89.

The ATET of going from treatment 0 to treatment 2 for those subjects who received treatment 2 is the mean of te_2 for those subjects for which $t = 2$, which is $(1.89 + 2.46)/2 \approx 2.18$. Receiving treatment 2 increased the outcome of those who received treatment 2 by an average of 2.18 relative to receiving the control.

Examples

In the remainder of this entry, we provide several examples demonstrating how to estimate multivalued treatments using `teffects`.

► Example 1: Potential outcomes with four treatment levels

`bdsianesi5.dta` contains an extract of data from [Blundell, Dearden, and Sianesi \(2005\)](#). In this dataset on individuals in the United Kingdom, `wages` records hourly wages in pounds; `ed` records the highest educational degree obtained; `paed` records the highest educational level obtained by each individual's father; `math7` records a score obtained on a standardized math test when the individual was seven; `read7` records a score obtained on a standardized reading test when the individual was seven; and `london` and `eastern` are indicators for whether an individual lives in the expensive area of London or the east. We want to know how the level of education obtained affects a person's wage.

We begin by using `mean` to report the estimated means of wages over the four education levels. The value labels on `mean` are coded as `none` for no degree, `O` for an O-level degree, `A` for an A-level degree, or `H` for a higher-education degree.

```
. use http://www.stata-press.com/data/r13/bdsianesi5
(Excerpt from Blundell, Dearden, & Sianesi (2005) JRSSA 168: 473)
. mean wage, over(ed)
Mean estimation                               Number of obs   =    1693
      none: ed = none
      O:   ed = 0
      A:   ed = A
      H:   ed = H
```

Over	Mean	Std. Err.	[95% Conf. Interval]	
wage				
none	6.057816	.154332	5.755114	6.360518
O	7.501648	.1807359	7.147158	7.856137
A	8.220637	.1540359	7.918516	8.522758
H	10.87703	.2257888	10.43417	11.31988

The output reveals that the estimated mean wage increases as the education level goes from no degree to an O-level degree, to an A-level degree, and to a higher-education degree, as we would expect. Once we control for other characteristics of each individual, do we still observe a positive effect of education on wage?

We use `teffects ra` (see [TE] [teffects ra](#)) to estimate the ATEs of the different education levels by regression adjustment (RA), controlling for each person's location, math score, and father's education level:

```
. teffects ra (wage london eastern paed math7, poisson) (ed)
Iteration 0:  EE criterion = 1.865e-18
Iteration 1:  EE criterion = 4.669e-30
Treatment-effects estimation           Number of obs   =    1693
Estimator      : regression adjustment
Outcome model  : Poisson
Treatment model: none
```

wage	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
ATE						
ed						
(O vs none)	1.181543	.3520371	3.36	0.001	.4915626	1.871523
(A vs none)	1.743079	.3282152	5.31	0.000	1.099789	2.386369
(H vs none)	3.972829	.3840024	10.35	0.000	3.220199	4.72546
P0mean						
ed						
none	6.525873	.2931933	22.26	0.000	5.951224	7.100521

Because wages are necessarily positive, we used the `poisson` option inside the outcome-model specification. The estimated POM of the control level of no degree is 6.53 pounds per hour. The estimated ATE of going from no degree to an O-level degree is 1.18 pounds per hour; the estimated ATE of going from no degree to an A-level degree is 1.74 pounds per hour; and the estimated ATE of going from no degree to a higher-education degree is 3.97 pounds per hour. All of these effects are highly significant.

For comparison purposes, we also use `teffects aipw` (see [TE] [teffects aipw](#)). We use the same outcome model as before. We use a multinomial logit model to predict education level, using math and reading scores and both the father's and the mother's educational attainment levels as predictors:

```

. teffects aipw (wage london eastern paed math7, poisson)
> (ed math7 read7 maed paed)
Iteration 0: EE criterion = 1.877e-18
Iteration 1: EE criterion = 7.691e-31
Treatment-effects estimation          Number of obs    =    1693
Estimator      : augmented IPW
Outcome model  : Poisson by ML
Treatment model: (multinomial) logit

```

wage	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
ATE						
ed						
(0 vs none)	1.748197	.3911167	4.47	0.000	.9816221	2.514771
(A vs none)	2.363228	.3741584	6.32	0.000	1.629891	3.096565
(H vs none)	4.359777	.4133059	10.55	0.000	3.549712	5.169842
POmean						
ed						
none	5.946184	.3391531	17.53	0.000	5.281456	6.610912

The results indicate slightly higher treatment effects relative to those indicated by `teffects ra`. That is largely because the AIPW estimator predicts a lower no-higher-education POM than the RA estimator.

◀

▶ Example 2: Expressing ATEs as percentages

As in the binary-treatment case, expressing the ATEs as percentages of the POM for the control level often aids interpretation. Here we first use the replay facility of `teffects aipw` along with the `coeflegend` option to see how the parameters are named.

```

. teffects, coeflegend
Treatment-effects estimation          Number of obs    =    1693
Estimator      : augmented IPW
Outcome model  : Poisson by ML
Treatment model: (multinomial) logit

```

wage	Coef.	Legend
ATE		
ed		
(0 vs none)	1.748197	_b[ATE:r1vs0.ed]
(A vs none)	2.363228	_b[ATE:r2vs0.ed]
(H vs none)	4.359777	_b[ATE:r3vs0.ed]
POmean		
ed		
none	5.946184	_b[POmean:r0.ed]

Now that we know the names, we can use `nlcom` to obtain the ATEs relative to the base-level POM:

```
. nlcom (_b[ATE:r1vs0.ed] / _b[P0mean:r0.ed])
>      (_b[ATE:r2vs0.ed] / _b[P0mean:r0.ed])
>      (_b[ATE:r3vs0.ed] / _b[P0mean:r0.ed])
      _nl_1:  _b[ATE:r1vs0.ed] / _b[P0mean:r0.ed]
      _nl_2:  _b[ATE:r2vs0.ed] / _b[P0mean:r0.ed]
      _nl_3:  _b[ATE:r3vs0.ed] / _b[P0mean:r0.ed]
```

wage	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
_nl_1	.2940031	.0808156	3.64	0.000	.1356075	.4523988
_nl_2	.3974361	.0840545	4.73	0.000	.2326923	.5621799
_nl_3	.7332059	.1068848	6.86	0.000	.5237156	.9426962

We see that going from no degree to an O-level degree increases wages by 29%, going from no degree to an A-level degree increases wages by 40%, and so on.

Although impressive, these changes are not presented in the way that is most commonly discussed. (There is a large amount of literature on the treatment effect of getting a higher-education degree.) In particular, we might rather want to know the percentage changes in wages relative to a person with an A-level degree. Next we estimate the ATEs treating an A-level degree as the control level; to do that, we use the `control()` option. We also specify `coeflegend` again because we are more interested in how the parameters are named rather than in their standard errors at this point:

```
. teffects aipw (wage london eastern paed math7, poisson)
> (ed math7 read7 maed paed), control(A) coeflegend

Iteration 0:  EE criterion = 1.870e-18
Iteration 1:  EE criterion = 1.934e-30

Treatment-effects estimation                Number of obs    =    1693
Estimator      : augmented IPW
Outcome model  : Poisson by ML
Treatment model: (multinomial) logit
```

wage	Coef.	Legend
ATE		
ed		
(none vs A)	-2.363228	_b[ATE:r0vs2.ed]
(O vs A)	-.6150312	_b[ATE:r1vs2.ed]
(H vs A)	1.996549	_b[ATE:r3vs2.ed]
P0mean		
ed		
A	8.309412	_b[P0mean:r2.ed]

Now we use `nlcom` to obtain the ATE of obtaining a higher-education degree as a percentage of the expected A-level wage:

```
. nlcom _b[ATE:r3vs2.ed] / _b[P0mean:r2.ed], noheader
```

wage	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
_nl_1	.2402756	.0355404	6.76	0.000	.1706177	.3099335

Having a higher-education degree is expected to increase a person's wage by 24% relative to someone with only an A-level degree.

▷ Example 3: Obtaining ATETs

In the previous example, we showed that on average, a higher-education degree increases a person's wage by 24% relative to someone with only an A-level degree. Sometimes, though, we would rather know how much the higher-education degree increases wages among the people who actually have a higher-education degree. To answer that question, we want to examine the ATET rather than the ATE.

Here we use the IPWRA estimator to obtain our answer. We specify the `control(A)` option so that an A-level education is treated as the basis for comparisons. We specify the `atet` option to obtain ATETs rather than ATEs, and we specify the `tlevel(H)` option to indicate that we want the ATETs to be calculated for the subset of people who actually receive higher-education degrees.

```
. teffects ipwra (wage london eastern paed math7, poisson)
> (ed math7 read7 maed paed), atet control(A) tlevel(H)

Iteration 0:   EE criterion = 2.731e-18
Iteration 1:   EE criterion = 5.078e-31

Treatment-effects estimation      Number of obs   =      1693
Estimator       : IPW regression adjustment
Outcome model   : Poisson
Treatment model : (multinomial) logit
```

	wage	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]
ATET						
	ed					
	(none vs A)	-2.87423	.361093	-7.96	0.000	-3.58196 -2.166501
	(O vs A)	-.8246604	.3609131	-2.28	0.022	-1.532037 -.1172837
	(H vs A)	1.866757	.3277701	5.70	0.000	1.224339 2.509174
P0mean						
	ed					
	A	9.010271	.2503971	35.98	0.000	8.519501 9.50104

The point estimates are similar to the ATEs we obtained above, suggesting that the means of the covariates among those with a higher-education degree are similar to the means for the entire population.

In output not shown to save space, we replayed the previous results with the `coeflegend` option to determine how the parameters are named. Armed with that information, we call `nlcom`:

```
. nlcom _b[ATET:r3vs2.ed] / _b[P0mean:r2.ed], noheader
```

	wage	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
	_nl_1	.207181	.0407528	5.08	0.000	.127307 .287055

Our estimate of the percentage increase is now noticeably smaller once we restrict ourselves to only those people who actually received a higher-education degree. However, because of the width of the confidence intervals, there is no evidence to suggest that the difference between the estimates is statistically significant.

▷ Example 4: ATEs comparing adjacent treatments

In the [first example](#), we obtained the three ATEs, and they were all expressed relative to the base level of no degree. Now we show how we can express the gains to an O-level degree relative to no degree, the gains to an A-level degree relative to an O-level degree, and the gains to a higher-education degree relative to an A-level degree.

First, we use an AIPW estimator to obtain all the POMs for our example dataset:

```
. teffects aipw (wage london eastern paed math7, poisson)
> (ed math7 read7 maed paed), pom
Iteration 0:   EE criterion = 1.877e-18
Iteration 1:   EE criterion = 1.436e-30
Treatment-effects estimation           Number of obs   =       1693
Estimator      : augmented IPW
Outcome model  : Poisson by ML
Treatment model: (multinomial) logit
```

wage	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
POMeans						
ed						
none	5.946184	.3391531	17.53	0.000	5.281456	6.610912
O	7.694381	.1915192	40.18	0.000	7.31901	8.069752
A	8.309412	.1563348	53.15	0.000	8.003001	8.615823
H	10.30596	.2285837	45.09	0.000	9.857945	10.75398

ATEs are contrasts of POMs, and here we show how to use `contrast` to obtain the estimated ATEs:

```
. contrast r.ed, nowald
Warning: cannot perform check for estimable functions.
Contrasts of marginal linear predictions
Margins      : asbalanced
```

	Contrast	Std. Err.	[95% Conf. Interval]	
POMeans				
ed				
(O vs none)	1.748197	.3911167	.9816221	2.514771
(A vs none)	2.363228	.3741584	1.629891	3.096565
(H vs none)	4.359777	.4133059	3.549712	5.169842

These estimated ATEs match those we obtained in [example 2](#).

Now that we know how to use `contrast` to obtain the ATEs based on the POMs, we can take advantage of `contrast`'s ability to obtain “reverse adjacent” contrasts, which compare each level with the previous level. We use the `ar.` operator with `contrast` to accomplish this:

```
. contrast ar.ed, nowald
Warning: cannot perform check for estimable functions.
Contrasts of marginal linear predictions
Margins      : asbalanced
```

	Contrast	Std. Err.	[95% Conf. Interval]	
P0means				
ed				
(O vs none)	1.748197	.3911167	.9816221	2.514771
(A vs O)	.6150312	.2432806	.13821	1.091852
(H vs A)	1.996549	.2730712	1.461339	2.531759

The results indicate that going from no degree to an O-level degree and going from an A-level degree to a higher-education degree have much larger effects than going from an O-level degree to an A-level degree.

◀

□ Technical note

The multivalued treatment AIPW estimators implemented in `teffects aipw` are EIF estimators based on the results of Cattaneo (2010). The results in Cattaneo (2010) are for semiparametric estimators, and we implement parametric versions. Of more practical importance, Cattaneo (2010) contains results for quantile treatment effects that are not implemented in `teffects` but implemented in the user-written `poparms` command discussed in Cattaneo, Drukker, and Holland (2013). See Emsley, Lunt, Pickles, and Dunn (2008) for another implementation of the AIPW estimator, and see Frölich and Melly (2010) for other commands that estimate quantile treatment effects.

□

References

- Abadie, A., and G. W. Imbens. 2006. Large sample properties of matching estimators for average treatment effects. *Econometrica* 74: 235–267.
- . 2012. Matching on the estimated propensity score. Harvard University and National Bureau of Economic Research. <http://www.hks.harvard.edu/fs/aabadie/pscore.pdf>.
- Blundell, R., L. Dearden, and B. Sianesi. 2005. Evaluating the effect of education on earnings: Models, methods and results from the National Child Development Survey. *Journal of the Royal Statistical Society, Series A* 168: 473–512.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154.
- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.
- Emsley, R., M. Lunt, A. Pickles, and G. Dunn. 2008. Implementing double-robust estimators of causal effects. *Stata Journal* 8: 334–353.
- Frölich, M., and B. Melly. 2010. Estimation of quantile treatment effects with Stata. *Stata Journal* 10: 423–457.
- Imbens, G. W. 2000. The role of the propensity score in estimating dose-response functions. *Biometrika* 87: 706–710.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

[TE] **teffects** — Treatment-effects estimation for observational data

[TE] **teffects intro** — Introduction to treatment effects for observational data

[TE] **teffects intro advanced** — Advanced introduction to treatment effects for observational data

Title

teffects nnmatch — Nearest-neighbor matching

Syntax

Remarks and examples
Also see

Menu

Stored results

Description

Methods and formulas

Options

References

Syntax

```
teffects nnmatch (ovar omvarlist) (tvar) [if] [in] [weight]  
[ , stat options ]
```

<i>stat</i>	Description
-------------	-------------

Stat	
<u>a</u> te	estimate average treatment effect in population; the default
<u>a</u> tet	estimate average treatment effect on the treated

<i>options</i>	Description
----------------	-------------

Model	
<u>n</u> neighbor(<i>#</i>)	specify number of matches per observation; default is <code>nneighbor(1)</code>
<u>b</u> iasadj(<i>varlist</i>)	correct for large-sample bias using specified variables
<u>e</u> match(<i>varlist</i>)	match exactly on specified variables

SE/Robust	
<u>v</u> ce(<i>vcetype</i>)	<i>vcetype</i> may be <code>vce(robust [, nn(<i>#</i>)])</code> ; use robust Abadie–Imbens standard errors with <i>#</i> matches <code>vce(iid)</code> ; use default Abadie–Imbens standard errors

Reporting	
<u>l</u> evel(<i>#</i>)	set confidence level; default is <code>level(95)</code>
<u>d</u> mvariables	display names of matching variables
<u>d</u> isplay_options	control column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

Advanced	
<u>c</u> aliper(<i>#</i>)	specify the maximum distance for which two observations are potential neighbors
<u>d</u> tolerance(<i>#</i>)	set maximum distance between individuals considered equal
<u>o</u> sample(<i>newvar</i>)	<i>newvar</i> identifies observations that violate the overlap assumption
<u>c</u> ontrol(<i>#</i> <i>label</i>)	specify the level of <i>tvar</i> that is the control
<u>t</u> level(<i>#</i> <i>label</i>)	specify the level of <i>tvar</i> that is the treatment
<u>g</u> enerate(<i>stub</i>)	generate variables containing the observation numbers of the nearest neighbors
<u>m</u> etric(<i>metric</i>)	select distance metric for covariates
<u>c</u> oeflegend	display legend instead of statistics

<i>metric</i>	Description
<code>mahalanobis</code>	inverse sample covariate covariance; the default
<code>ivariance</code>	inverse diagonal sample covariate covariance
<code>euclidean</code>	identity
<code>matrix matname</code>	user-supplied scaling matrix

tvar must contain integer values representing the treatment levels.

omvarlist may contain factor variables; see [U] 11.4.3 [Factor variables](#).

by and *statsby* are allowed; see [U] 11.1.10 [Prefix commands](#).

fweights are allowed; see [U] 11.1.6 [weight](#).

coeflegend does not appear in the dialog box.

See [U] 20 [Estimation and postestimation commands](#) for more capabilities of estimation commands.

Menu

Statistics > Treatment effects > Matching estimators > Nearest-neighbor matching

Description

`teffects nnmatch` estimates treatment effects from observational data by nearest-neighbor matching. NNM imputes the missing potential outcome for each subject by using an average of the outcomes of similar subjects that receive the other treatment level. Similarity between subjects is based on a weighted function of the covariates for each observation. The average treatment effect (ATE) is computed by taking the average of the difference between the observed and imputed potential outcomes for each subject.

Options

Model

`nneighbor(#)` specifies the number of matches per observation. The default is `nneighbor(1)`. Each observation is matched with at least the specified number of observations from the other treatment level. `nneighbor()` must specify an integer greater than or equal to 1 but no larger than the number of observations in the smallest treatment group.

`biasadj(varlist)` specifies that a linear function of the specified covariates be used to correct for a large-sample bias that exists when matching on more than one continuous covariate. By default, no correction is performed.

[Abadie and Imbens \(2006, 2011\)](#) show that nearest-neighbor matching estimators are not consistent when matching on two or more continuous covariates and propose a bias-corrected estimator that is consistent. The correction term uses a linear function of variables specified in `biasadj()`; see [example 3](#).

`ematch(varlist)` specifies that the variables in *varlist* match exactly. All variables in *varlist* must be numeric and may be specified as factors. `teffects nnmatch` exits with an error if any observations do not have the requested exact match.

Stat

stat is one of two statistics: `ate` or `atet`. `ate` is the default.

`ate` specifies that the average treatment effect be estimated.

`atet` specifies that the average treatment effect on the treated be estimated.

SE/Robust

`vce(vctype)` specifies the standard errors that are reported. By default, `teffects nnmatch` uses two matches in estimating the robust standard errors.

`vce(robust [, nn(#)])` specifies that robust standard errors be reported and that the requested number of matches be used optionally.

`vce(iid)` specifies that standard errors for independently and identically distributed data be reported.

The standard derivative-based standard-error estimators cannot be used by `teffects nnmatch`, because these matching estimators are not differentiable. The implemented methods were derived by [Abadie and Imbens \(2006, 2011, 2012\)](#); see *Methods and formulas*.

As discussed in [Abadie and Imbens \(2008\)](#), bootstrap estimators do not provide reliable standard errors for the estimator implemented by `teffects nnmatch`.

Reporting

`level(#)`; see [\[R\] estimation options](#).

`dmvariables` specifies that the matching variables be displayed.

display_options: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `no!stretch`; see [\[R\] estimation options](#).

Advanced

`caliper(#)` specifies the maximum distance at which two observations are a potential match. By default, all observations are potential matches regardless of how dissimilar they are.

The distance is based on `omvarlist`. If an observation does not have at least `nneighbor(#)` matches, `teffects nnmatch` exits with an error message. Use option `osample(newvar)` to identify all observations that are deficient in matches.

`dtolerance(#)` specifies the tolerance used to determine exact matches. The default value is `dtolerance(sqrt(c(epsdouble)))`.

Integer-valued variables are usually used for exact matching. The `dtolerance()` option is useful when continuous variables are used for exact matching.

`osample(newvar)` specifies that indicator variable `newvar` be created to identify observations that violate the overlap assumption. This variable will identify all observations that do not have at least `nneighbor(#)` matches in the opposite treatment group within `caliper(#)` (for `metric()` distance matching) or `dtolerance(#)` (for `ematch(varlist)` exact matches).

The `vce(robust, nn(#))` option also requires at least `#` matches in the same treatment group within the distance specified by `caliper(#)` or within the exact matches specified by `dtolerance(#)`.

The average treatment effect on the treated, option `atet`, using `vce(iid)` requires only `nneighbor(#)` control group matches for the treated group.

`control(# | label)` specifies the level of `tvar` that is the control. The default is the first treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(# | label)` specifies the level of `tvar` that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

`generate(stub)` specifies that the observation numbers of the nearest neighbors be stored in the new variables `stub1`, `stub2`, `...` This option is required if you wish to perform postestimation based on the matching results. The number of variables generated may be more than `nneighbors(#)` because of tied distances. These variables may not already exist.

`metric(metric)` specifies the distance matrix used as the weight matrix in a quadratic form that transforms the multiple distances into a single distance measure; see [Nearest-neighbor matching estimator](#) in *Methods and formulas* for details.

The following option is available with `teffects nmatch` but is not shown in the dialog box:

`coeflegend`; see [\[R\] estimation options](#).

Remarks and examples

The NNM method of treatment-effect estimation imputes the missing potential outcome for each individual by using an average of the outcomes of similar subjects that receive the other treatment level. Similarity between subjects is based on a weighted function of the covariates for each observation. The average treatment effect (ATE) is computed by taking the average of the difference between the observed and potential outcomes for each subject.

`teffects nmatch` determines the “nearest” by using a weighted function of the covariates for each observation. By default, the Mahalanobis distance is used, in which the weights are based on the inverse of the covariates’ variance–covariance matrix. `teffects nmatch` also allows you to request exact matching for categorical covariates. For example, you may want to force all matches to be of the same gender or race.

NNM is nonparametric in that no explicit functional form for either the outcome model or the treatment model is specified. This flexibility comes at a price; the estimator needs more data to get to the true value than an estimator that imposes a functional form. More formally, the NNM estimator converges to the true value at a rate slower than the parametric rate, which is the square root of the sample size, when matching on more than one continuous covariate. `teffects nmatch` uses bias correction to fix this problem. `teffects psmatch` implements an alternative to bias correction; the method matches on a single continuous covariate, the estimated treatment probabilities. See [\[TE\] teffects intro](#) or [\[TE\] teffects intro advanced](#) for more information about this estimator.

We will illustrate the use of `teffects nmatch` by using data from a study of the effect of a mother’s smoking status during pregnancy (`mbsmoke`) on infant birthweight (`bweight`) as reported by [Cattaneo \(2010\)](#). This dataset also contains information about each mother’s age (`mage`), education level (`medu`), marital status (`mmarried`), whether the first prenatal exam occurred in the first trimester (`prenatal1`), whether this baby was the mother’s first birth (`fbaby`), and the father’s age (`fage`).

▷ Example 1: Estimating the ATE

We begin by using `teffects nmatch` to estimate the average treatment effect of `mbsmoke` on `bweight`. Subjects are matched using the Mahalanobis distance defined by covariates `mage`, `prenatal1`, `mmarried`, and `fbaby`.

```
. use http://www.stata-press.com/data/r13/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects nnmatch (bweight mage prenatal1 mmarried fbaby) (mbsmoke)
Treatment-effects estimation      Number of obs      =      4642
Estimator      : nearest-neighbor matching      Matches: requested =      1
Outcome model  : matching                        min =            1
Distance metric: Mahalanobis                    max =           139
```

	bweight	Coef.	AI Robust Std. Err.	z	P> z	[95% Conf. Interval]
ATE	mbsmoke (smoker vs nonsmoker)	-240.3306	28.43006	-8.45	0.000	-296.0525 -184.6087

Smoking causes infants' birthweights to be reduced by an average of 240 grams. ◀

When the model includes indicator and categorical variables, you may want to restrict matches to only those subjects who are in the same category. The `ematch()` option of `teffects nnmatch` allows you to specify such variables that must match exactly.

▷ Example 2: Exact matching

Here we use the `ematch()` option to require exact matches on the binary variables `prenatal1`, `mmarried`, and `fbaby`. We also use Euclidean distance, rather than the default Mahalanobis distance, to match on the continuous variable `mage`, which uses Euclidean distance.

```
. teffects nnmatch (bweight mage) (mbsmoke),
> ematch(prenatal1 mmarried fbaby) metric(euclidean)
Treatment-effects estimation      Number of obs      =      4642
Estimator      : nearest-neighbor matching      Matches: requested =      1
Outcome model  : matching                        min =            1
Distance metric: Euclidean                    max =           139
```

	bweight	Coef.	AI Robust Std. Err.	z	P> z	[95% Conf. Interval]
ATE	mbsmoke (smoker vs nonsmoker)	-240.3306	28.43006	-8.45	0.000	-296.0525 -184.6087

Abadie and Imbens (2006, 2011) have shown that nearest-neighbor matching estimators are not consistent when matching on two or more continuous covariates. A bias-corrected estimator that uses a linear function of variables can be specified with `biasadj()`. ◀

▷ Example 3: Bias adjustment

Here we match on two continuous variables, `mage` and `fage`, and we use the bias-adjusted estimator:

```
. teffects nnmatch (bweight mage fage) (mbsmoke),
> ematch(prenatal1 mmarried fbaby) biasadj(mage fage)

Treatment-effects estimation      Number of obs      =      4642
Estimator      : nearest-neighbor matching      Matches: requested =      1
Outcome model  : matching                        min =              1
Distance metric: Mahalanobis                    max =              25
```

bweight	Coef.	AI Robust Std. Err.	z	P> z	[95% Conf. Interval]	
ATE						
mbsmoke (smoker vs nonsmoker)	-223.8389	26.19973	-8.54	0.000	-275.1894	-172.4883

◀

Stored results

`teffects nnmatch` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level j
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(k_nneighbor)</code>	requested number of matches
<code>e(k_nnmin)</code>	minimum number of matches
<code>e(k_nnmax)</code>	maximum number of matches
<code>e(k_robust)</code>	matches for robust VCE

Macros

<code>e(cmd)</code>	<code>teffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(emvarlist)</code>	exact match variables
<code>e(bavarlist)</code>	variables used in bias adjustment
<code>e(mvarlist)</code>	match variables
<code>e(subcmd)</code>	<code>nnmatch</code>
<code>e(metric)</code>	<code>mahalanobis</code> , <code>ivariance</code> , <code>euclidean</code> , or matrix <code>matname</code>
<code>e(stat)</code>	statistic estimated, <code>ate</code> or <code>atet</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<code>vcetype</code> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. Err.
<code>e(datasignature)</code>	the checksum
<code>e(datasignaturevars)</code>	variables used in calculation of checksum
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>

Matrices	
e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators
Functions	
e(sample)	marks estimation sample

Methods and formulas

The methods and formulas presented here provide the technical details underlying the estimators implemented in `teffects nnmatch` and `teffects psmatch`. See *Methods and formulas* of [TE] [teffects aipw](#) for the methods and formulas used by `teffects aipw`, `teffects ipw`, `teffects ipwra`, and `teffects ra`.

Methods and formulas are presented under the following headings:

Nearest-neighbor matching estimator
Bias-corrected matching estimator
Propensity-score matching estimator
PSM, ATE, and ATET variance adjustment

Nearest-neighbor matching estimator

`teffects nnmatch` implements the nearest-neighbor matching (NNM) estimator for the average treatment effect (ATE) and the average treatment effect on the treated (ATET). This estimator was derived by [Abadie and Imbens \(2006, 2011\)](#) and was previously implemented in Stata as discussed in [Abadie et al. \(2004\)](#).

`teffects psmatch` implements nearest-neighbor matching on an estimated propensity score. A propensity score is a conditional probability of treatment. The standard errors implemented in `teffects psmatch` were derived by [Abadie and Imbens \(2012\)](#).

`teffects nnmatch` and `teffects psmatch` permit two treatment levels: the treatment group with $t = 1$ and a control group with $t = 0$.

Matching estimators are based on the potential-outcome model, in which each individual has a well-defined outcome for each treatment level; see [TE] [teffects intro](#). In the binary-treatment potential-outcome model, y_1 is the potential outcome obtained by an individual if given treatment-level 1 and y_0 is the potential outcome obtained by each individual i if given treatment-level 0. The problem posed by the potential-outcome model is that only y_{1i} or y_{0i} is observed, never both. y_{0i} and y_{1i} are realizations of the random variables y_0 and y_1 . Throughout this document, i subscripts denote realizations of the corresponding, unsubscripted random variables.

Formally, the ATE is

$$\tau_1 = E(y_1 - y_0)$$

and the ATET is

$$\delta_1 = E(y_1 - y_0 | t = 1)$$

These expressions imply that we must have some solution to the missing-data problem that arises because we only observe either y_{1i} or y_{0i} , not both.

For each individual, NNM uses an average of the individuals that are most similar, but get the other treatment level, to predict the unobserved potential outcome. NNM uses the covariates $\{x_1, x_2, \dots, x_p\}$ to find the most similar individuals that get the other treatment level.

More formally, consider the vector of covariates $\mathbf{x}_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,p}\}$ and frequency weight w_i for observation i . The distance between \mathbf{x}_i and \mathbf{x}_j is parameterized by the vector norm

$$\|\mathbf{x}_i - \mathbf{x}_j\|_S = \{(\mathbf{x}_i - \mathbf{x}_j)' \mathbf{S}^{-1} (\mathbf{x}_i - \mathbf{x}_j)\}^{1/2}$$

where \mathbf{S} is a given symmetric, positive-definite matrix.

Using this distance definition, we find that the set of nearest-neighbor indices for observation i is

$$\Omega_m^{\mathbf{x}}(i) = \{j_1, j_2, \dots, j_{m_i} \mid t_{j_k} = 1 - t_i, \|\mathbf{x}_i - \mathbf{x}_{j_k}\|_S < \|\mathbf{x}_i - \mathbf{x}_l\|_S, t_l = 1 - t_i, l \neq j_k\}$$

Here m_i is the smallest number such that the number of elements in each set, $m_i = |\Omega_m^{\mathbf{x}}(i)| = \sum_{j \in \Omega_m^{\mathbf{x}}(i)} w_j$, is at least m , the desired number of matches. You set the size of m using the `nneighbors(#)` option. The number of matches for the i th observation may not equal m because of ties or if there are not enough observations with a distance from observation i within the caliper limit, c , $\|\mathbf{x}_i - \mathbf{x}_j\|_S \leq c$. You may set the caliper limit by using the `caliper(#)` option. For ease of notation, we will use the abbreviation $\Omega(i) = \Omega_m^{\mathbf{x}}(i)$.

With the `metric(string)` option, you have three choices for the scaling matrix \mathbf{S} : Mahalanobis, inverse variance, or Euclidean.

$$\mathbf{S} = \begin{cases} \frac{(\mathbf{X} - \bar{\mathbf{x}}' \mathbf{1}_n)' \mathbf{W} (\mathbf{X} - \bar{\mathbf{x}}' \mathbf{1}_n)}{\sum_i w_i - 1} & \text{if } \text{metric} = \text{mahalanobis} \\ \text{diag} \left\{ \frac{(\mathbf{X} - \bar{\mathbf{x}}' \mathbf{1}_n)' \mathbf{W} (\mathbf{X} - \bar{\mathbf{x}}' \mathbf{1}_n)}{\sum_i w_i - 1} \right\} & \text{if } \text{metric} = \text{ivariance} \\ \mathbf{I}_p & \text{if } \text{metric} = \text{euclidean} \end{cases}$$

where $\mathbf{1}_n$ is an $n \times 1$ vector of ones, \mathbf{I}_p is the identity matrix of order p , $\bar{\mathbf{x}} = (\sum_i w_i \mathbf{x}_i) / (\sum_i w_i)$, and \mathbf{W} is an $n \times n$ diagonal matrix containing frequency weights.

The NNM method predicting the potential outcome for the i th observation as a function of the observed y_i is

$$\hat{y}_{ti} = \begin{cases} y_i & \text{if } t_i = t \\ \frac{\sum_{j \in \Omega(i)} w_j y_j}{\sum_{j \in \Omega(i)} w_j} & \text{otherwise} \end{cases}$$

for $t \in \{0, 1\}$.

We are now set to provide formulas for estimates $\hat{\tau}_1$, the ATE, and $\hat{\delta}_1$, the ATET,

$$\hat{\tau}_1 = \frac{\sum_{i=1}^n w_i (\hat{y}_{1i} - \hat{y}_{0i})}{\sum_{i=1}^n w_i} = \frac{\sum_{i=1}^n w_i (2t_i - 1) \{1 + K_m(i)\} y_i}{\sum_{i=1}^n w_i}$$

$$\hat{\delta}_1 = \frac{\sum_{i=1}^n t_i w_i (\hat{y}_{1i} - \hat{y}_{0i})}{\sum_{i=1}^n t_i w_i} = \frac{\sum_{i=1}^n \{t_i - (1 - t_i) K_m(i)\} y_i}{\sum_{i=1}^n t_i w_i}$$

where

$$K_m(i) = \sum_{j=1}^n I\{i \in \Omega(j)\} \frac{w_j}{\sum_{k \in \Omega(j)} w_k}$$

The estimated variance of $\widehat{\tau}_1$ and $\widehat{\delta}_1$ are computed as

$$\widehat{\sigma}_\tau^2 = \frac{\sum_{i=1}^n w_i \left[(\widehat{y}_{1i} - \widehat{y}_{0i} - \widehat{\tau}_1)^2 + \widehat{\xi}_i^2 \{K_m^2(i) + 2K_m(i) - K'_m(i)\} \right]}{\left(\sum_{i=1}^n w_i\right)^2}$$

$$\widehat{\sigma}_\delta^2 = \frac{\sum_{i=1}^n t_i w_i \left[(\widehat{y}_{1i} - \widehat{y}_{0i} - \widehat{\delta}_1)^2 + \widehat{\xi}_i^2 \{K_m^2(i) - K'_m(i)\} \right]}{\left(\sum_{i=1}^n t_i w_i\right)^2}$$

where

$$K'_m(i) = \sum_{j=1}^n I\{i \in \Omega(j)\} \frac{w_j}{\left(\sum_{k \in \Omega(j)} w_k\right)^2}$$

and $\xi_i^2 = \text{var}(y_{ti} | \mathbf{x}_i)$ is the conditional outcome variance. If we can assume that ξ_i^2 does not vary with the covariates or treatment (homoskedastic), then we can compute an ATE estimate of ξ_τ^2 as

$$\widehat{\xi}_\tau^2 = \frac{1}{2 \sum_i^n w_i} \sum_{i=1}^n w_i \left[\frac{\sum_{j \in \Omega(i)} w_j \{y_i - y_j(1 - t_i) - \widehat{\tau}_1\}^2}{\sum_{j \in \Omega(i)} w_j} \right]$$

and an ATET estimate of ξ_δ^2 as

$$\widehat{\xi}_\delta^2 = \frac{1}{2 \sum_i^n t_i w_i} \sum_{i=1}^n t_i w_i \left[\frac{\sum_{j \in \Omega(i)} t_j w_j \{y_i - y_j(1 - t_i) - \widehat{\delta}_1\}^2}{\sum_{j \in \Omega(i)} t_j w_j} \right]$$

If the conditional outcome variance is dependent on the covariates or treatment, we require an estimate for ξ_i^2 at each observation. In this case, we require a second matching procedure, where we match on observations within the same treatment group.

Define the within-treatment matching set

$$\Psi_h^{\mathbf{x}}(i) = \{j_1, j_2, \dots, j_{h_i} \mid t_{j_k} = t_i, \|\mathbf{x}_i - \mathbf{x}_{j_k}\|_S < \|\mathbf{x}_i - \mathbf{x}_l\|_S, t_l = t_i, l \neq j_k\}$$

where h is the desired set size. As before, the number of elements in each set, $h_i = |\Psi_h^{\mathbf{x}}(i)|$, may vary depending on ties and the value of the caliper. You set h using the `vce(robust, nn(#))` option. As before, we will use the abbreviation $\Psi(i) = \Psi_h^{\mathbf{x}}(i)$ where convenient.

We estimate $\xi_{t_i}^2$ by

$$\widehat{\xi}_{t_i}^2(\mathbf{x}_i) = \frac{\sum_{j \in \Psi(i)} w_j (y_j - \bar{y}_{\Psi(i)})^2}{\sum_{j \in \Psi(i)} w_j - 1} \quad \text{where} \quad \bar{y}_{\Psi(i)} = \frac{\sum_{j \in \Psi(i)} w_j y_j}{\sum_{j \in \Psi(i)} w_j - 1}$$

Bias-corrected matching estimator

When matching on more than one continuous covariate, the matching estimator described above is biased, even in infinitely large samples; in other words, it is not \sqrt{n} -consistent; see [Abadie and Imbens \(2006, 2011\)](#). Following [Abadie and Imbens \(2011\)](#) and [Abadie et al. \(2004\)](#), **teffects nmatch** makes an adjustment based on the regression functions $\mu_t(\tilde{\mathbf{x}}_i) = E(y_t \mid \tilde{\mathbf{x}} = \tilde{\mathbf{x}}_i)$, for $t = 0, 1$ and the set of covariates $\tilde{\mathbf{x}}_i = (\tilde{x}_{i,1}, \dots, \tilde{x}_{i,q})$. The bias-correction covariates may be the same as the NNM covariates \mathbf{x}_i . We denote the least-squares estimates as $\hat{\mu}_t(\tilde{\mathbf{x}}_i) = \hat{\nu}_t + \hat{\beta}'_t \tilde{\mathbf{x}}_i$, where we regress $\{y_i \mid t_i = t\}$ onto $\{\tilde{\mathbf{x}}_i \mid t_i = t\}$ with weights $w_i K_m(i)$, for $t = 0, 1$.

Given the estimated regression functions, the bias-corrected predictions for the potential outcomes are computed as

$$\hat{y}_{ti} = \begin{cases} y_i & \text{if } t_i = t \\ \frac{\sum_{j \in \Omega_m^{\mathbf{x}}(i)} w_j \{y_j + \hat{\mu}_t(\tilde{\mathbf{x}}_i) - \hat{\mu}_t(\tilde{\mathbf{x}}_j)\}}{\sum_{j \in \Omega_m^{\mathbf{x}}(i)} w_j - 1} & \text{otherwise} \end{cases}$$

The `biasadj(varlist)` option specifies the bias-adjustment covariates $\tilde{\mathbf{x}}_i$.

Propensity-score matching estimator

The propensity-score matching (PSM) estimator uses a treatment model (TM), $p(\mathbf{z}_i, t, \gamma)$, to model the conditional probability that observation i receives treatment t given covariates \mathbf{z}_i . The literature calls $p(\mathbf{z}_i, t, \gamma)$ a propensity score, and PSM matches on the estimated propensity scores.

When matching on the estimated propensity score, the set of nearest-neighbor indices for observation i , $i = 1, \dots, n$, is

$$\Omega_m^{\mathbf{p}}(i) = \{j_1, j_2, \dots, j_{m_i} \mid t_{j_k} = 1 - t_i, |\hat{p}_i(t) - \hat{p}_{j_k}(t)| < |\hat{p}_i(t) - \hat{p}_l(t)|, t_l = 1 - t_i, l \neq j_k\}$$

where $\hat{p}_i(t) = p(\mathbf{z}_i, t, \hat{\gamma})$. As was the case with the NNM estimator, m_i is the smallest number such that the number of elements in each set, $m_i = |\Omega_m^{\mathbf{p}}(i)| = \sum_{j \in \Omega_m^{\mathbf{p}}(i)} w_j$, is at least m , the desired number of matches, set by the `nneighbors(#)` option.

We define the within-treatment matching set analogously,

$$\Psi_h^{\mathbf{p}}(i) = \{j_1, j_2, \dots, j_{h_i} \mid t_{j_k} = t_i, |\hat{p}_i(t) - \hat{p}_{j_k}(t)| < |\hat{p}_i(t) - \hat{p}_l(t)|, t_l = t_i, l \neq j_k\}$$

where h is the desired number of within-treatment matches, and $h_i = |\Psi_h^{\mathbf{p}}(i)|$, for $i = 1, \dots, n$, may vary depending on ties and the value of the caliper. The sets $\Psi_h^{\mathbf{p}}(i)$ are required to compute standard errors for $\hat{\tau}_1$ and $\hat{\delta}_1$.

Once a matching set is computed for each observation, the potential-outcome mean, ATE, and ATET computations are identical to those of NNM. The ATE and ATET standard errors, however, must be adjusted because the TM parameters were estimated; see [Abadie and Imbens \(2012\)](#).

PSM, ATE, and ATET variance adjustment

The variances for $\hat{\tau}_1$ and $\hat{\delta}_1$ must be adjusted because we use $\hat{\gamma}$ instead of γ . The adjusted variances for $\hat{\tau}_1$ and $\hat{\delta}_1$ have the following forms, respectively:

$$\begin{aligned}\hat{\sigma}_{\tau,\text{adj}}^2 &= \hat{\sigma}_{\tau}^2 + \hat{\mathbf{c}}_{\tau}' \hat{\mathbf{V}}_{\gamma} \hat{\mathbf{c}}_{\tau} \\ \hat{\sigma}_{\delta,\text{adj}}^2 &= \hat{\sigma}_{\delta}^2 - \hat{\mathbf{c}}_{\delta}' \hat{\mathbf{V}}_{\gamma} \hat{\mathbf{c}}_{\delta} + \frac{\partial \hat{\delta}_1}{\partial \gamma'} \hat{\mathbf{V}}_{\gamma} \frac{\partial \hat{\delta}_1}{\partial \gamma}\end{aligned}$$

In both equations, the matrix $\hat{\mathbf{V}}_{\gamma}$ is the TM coefficient variance–covariance matrix.

The adjustment term for ATE can be expressed as

$$\hat{\mathbf{c}}_{\tau} = \frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^n w_i f(\mathbf{z}'_i \hat{\gamma}) \left(\frac{\widehat{\text{cov}}(\mathbf{z}_i, \hat{y}_{i1})}{\hat{p}_i(1)} + \frac{\widehat{\text{cov}}(\mathbf{z}_i, \hat{y}_{i0})}{\hat{p}_i(0)} \right)$$

where

$$f(\mathbf{z}'_i \hat{\gamma}) = \frac{d p(\mathbf{z}_i, 1, \hat{\gamma})}{d(\mathbf{z}'_i \hat{\gamma})}$$

is the derivative of $p(\mathbf{z}_i, 1, \hat{\gamma})$ with respect to $\mathbf{z}'_i \hat{\gamma}$, and

$$\widehat{\text{cov}}(\mathbf{z}_i, \hat{y}_{ti}) = \begin{cases} \frac{\sum_{j \in \Psi_h(i)} w_j (\mathbf{z}_j - \bar{\mathbf{z}}_{\Psi_h(i)}) (y_j - \bar{y}_{\Psi_h(i)})}{\sum_{j \in \Psi_h(i)} w_j - 1} & \text{if } t_i = t \\ \frac{\sum_{j \in \Omega_h(i)} w_j (\mathbf{z}_j - \bar{\mathbf{z}}_{\Omega_h(i)}) (y_j - \bar{y}_{\Omega_h(i)})}{\sum_{j \in \Omega_h(i)} w_j - 1} & \text{otherwise} \end{cases}$$

is a $p \times 1$ vector with

$$\bar{\mathbf{z}}_{\Psi_h(i)} = \frac{\sum_{j \in \Psi_h(i)} w_j \mathbf{z}_j}{\sum_{j \in \Psi_h(i)} w_j} \quad \bar{\mathbf{z}}_{\Omega_h(i)} = \frac{\sum_{j \in \Omega_h(i)} w_j \mathbf{z}_j}{\sum_{j \in \Omega_h(i)} w_j} \quad \text{and} \quad \bar{y}_{\Omega_h(i)} = \frac{\sum_{j \in \Omega_h(i)} w_j y_j}{\sum_{j \in \Omega_h(i)} w_j}$$

Here we have used the notation $\Psi_h(i) = \Psi_h^p(i)$ and $\Omega_h(i) = \Omega_h^p(i)$ to stress that the within-treatment and opposite-treatment clusters used in computing $\hat{\sigma}_{\tau,\text{adj}}^2$ and $\hat{\sigma}_{\delta,\text{adj}}^2$ are based on h instead of the cluster $\Omega_m^p(i)$ based on m used to compute $\hat{\tau}_1$ and $\hat{\delta}_1$, although you may desire to have $h = m$.

The adjustment term \mathbf{c}_δ for the ATET estimate has two components, $\mathbf{c}_\delta = \mathbf{c}_{\delta,1} + \mathbf{c}_{\delta,2}$, defined as

$$\mathbf{c}_{\delta,1} = \frac{1}{\sum_{i=1}^n t_i w_i} \sum_{i=1}^n w_i \mathbf{z}_i f(\mathbf{z}'_i \hat{\boldsymbol{\gamma}}) \left(\tilde{y}_{1i} - \tilde{y}_{0i} - \hat{\delta}_1 \right)$$

$$\mathbf{c}_{\delta,2} = \frac{1}{\sum_{i=1}^n t_i w_i} \sum_{i=1}^n w_i f(\mathbf{z}'_i \hat{\boldsymbol{\gamma}}) \left\{ \widehat{\text{cov}}(\mathbf{z}_i, \hat{y}_{1i}) + \frac{\hat{p}_i(1)}{\hat{p}_i(0)} \widehat{\text{cov}}(\mathbf{z}_i, \hat{y}_{0i}) \right\}$$

where

$$\tilde{y}_{ti} = \begin{cases} \frac{\sum_{j \in \Psi_h(-i)} w_j y_j}{\sum_{j \in \Psi_h(-i)} w_j} & \text{if } t = t_i \\ \frac{\sum_{j \in \Omega_h} w_j y_j}{\sum_{j \in \Omega_h} w_j} & \text{otherwise} \end{cases}$$

and the within-treatment matching sets $\Psi_h(-i) = \Psi_h^p(-i)$ are similar to $\Psi_h^p(i)$ but exclude observation i :

$$\Psi_h^p(-i) = \{j_1, j_2, \dots, j_{h_i} \mid j_k \neq i, t_{j_k} = t_i, |\hat{p}_i - \hat{p}_{j_k}| < |\hat{p}_i - \hat{p}_l|, t_l = t_i, l \notin \{i, j_k\}\}$$

Finally, we cover the computation of $\frac{\partial \hat{\delta}_1}{\partial \boldsymbol{\gamma}}$ in the third term on the right-hand side of $\hat{\sigma}_{\delta, \text{adj}}^2$. Here we require yet another clustering, but we match on the opposite treatment by using the covariates $\mathbf{z}_i = (z_{i,1}, \dots, z_{i,p})'$. We will denote these cluster sets as $\Omega_m^{\mathbf{z}}(i)$, for $i = 1, \dots, n$.

The estimator of the $p \times 1$ vector $(\partial \delta_1) / (\partial \boldsymbol{\gamma})$ is computed as

$$\frac{\partial \hat{\delta}_1}{\partial \boldsymbol{\gamma}} = \frac{1}{\sum_{i=1}^n t_i w_i} \sum_{i=1}^n \mathbf{z}_i f(\mathbf{z}'_i \hat{\boldsymbol{\gamma}}) \left\{ (2t_i - 1)(y_i - \bar{y}_{\Omega_m^{\mathbf{z}}(i)}) - \hat{\delta}_1 \right\}$$

where

$$\bar{y}_{\Omega_m^{\mathbf{z}}(i)} = \frac{\sum_{j \in \Omega_m^{\mathbf{z}}(i)} w_j y_j}{\sum_{j \in \Omega_m^{\mathbf{z}}(i)} w_j}$$

References

- Abadie, A., D. M. Drukker, J. L. Herr, and G. W. Imbens. 2004. Implementing matching estimators for average treatment effects in Stata. *Stata Journal* 4: 290–311.
- Abadie, A., and G. W. Imbens. 2006. Large sample properties of matching estimators for average treatment effects. *Econometrica* 74: 235–267.
- . 2008. On the failure of the bootstrap for matching estimators. *Econometrica* 76: 1537–1557.
- . 2011. Bias-corrected matching estimators for average treatment effects. *Journal of Business and Economic Statistics* 29: 1–11.
- . 2012. Matching on the estimated propensity score. Harvard University and National Bureau of Economic Research. <http://www.hks.harvard.edu/fs/aabadie/pscore.pdf>.

Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154.

Also see

- [TE] **teffects postestimation** — Postestimation tools for teffects
- [TE] **teffects** — Treatment-effects estimation for observational data
- [TE] **teffects psmatch** — Propensity-score matching
- [U] **20 Estimation and postestimation commands**

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

Syntax

```
teffects overlap [ , treat_options kden_options ]
```

treat_options Description

Main

ptlevel(*treat_level*) calculate predicted probabilities for treatment level *treat_level*; by default, `ptlevel()` corresponds to the first treatment level

tlevels(*treatments*) specify conditioning treatment levels; default is all treatment levels

nolabel use treatment level values and not value labels in legend and axis titles

kden_options Description

Main

kernel(*kernel*) specify kernel function; default is `kernel(triangle)`

n(#) estimate densities using # points; default is `e(N)`, the number of observations in the estimation sample

bwidth(#) half-width of kernel

at(*var_x*) estimate densities using the values specified by *var_x*

Kernel plots

line#opts(*cline_options*) affect rendition of density for conditioning treatment #

Add plots

addplot(*plot*) add other plots to the generated graph

Y axis, X axis, Titles, Legend, Overall

twoway_options any options other than `by()` documented in [G-3] *twoway_options*

kernel Description

triangle triangle kernel function; the default

epanechnikov Epanechnikov kernel function

epan2 alternative Epanechnikov kernel function

biweight biweight kernel function

cosine cosine trace kernel function

gaussian Gaussian kernel function

parzen Parzen kernel function

rectangle rectangle kernel function

Menu

Statistics > Treatment effects > Overlap plots

Description

One of the assumptions required to use the `teffects` estimators is the overlap assumption, which states that each individual has a positive probability of receiving each treatment level. `teffects overlap`, a postestimation command, plots the estimated densities of the probability of getting each treatment level. These plots can be used to check whether the overlap assumption is violated.

Options

Main

`ptlevel(treat_level)` specifies that predicted probabilities be calculated for treatment level *treat_level*. The default is `ptlevel(first)`, where *first* is the first treatment level.

`tlevels(treatments)` specifies the observations for which to obtain predicted probabilities. By default, all treatment levels are used. Specify *treatments* as a space-delimited list.

For instance,

```
. teffects overlap, ptlevel(1) tlevels(1 2)
```

says to predict the probability of getting treatment level 1 for those subjects who actually obtained treatment levels 1 or 2.

`no label` specifies that treatment level values and not value labels be used in legend and axis titles.

`kernel(kernel)` specifies the kernel function for use in calculating the kernel density estimates. The default kernel is the triangle kernel (`triangle`).

`n(#)` specifies the number of points at which the density estimate is to be evaluated. The default is `e(N)`, the estimation sample size.

`bwidth(#)` specifies the half-width of the kernel, the width of the density window around each point. If `bwidth()` is not specified, the “optimal” width is calculated and used; see [R] [kdensity](#). The optimal width is the width that would minimize the mean integrated squared error if the data were Gaussian and a Gaussian kernel were used, so it is not optimal in any global sense. In fact, for multimodal and highly skewed densities, this width is usually too wide and oversmooths the density (Silverman 1992).

`at(var_x)` specifies a variable that contains the values at which the density should be estimated. This option allows you to more easily obtain density estimates for different variables or different subsamples of a variable and then overlay the estimated densities for comparison.

Kernel plots

`line#opts(cline_options)` affect the rendition of the plotted kernel density estimates. See [G-3] [cline_options](#).

Add plots

`addplot(plot)` provides a way to add other plots to the generated graph. See [G-3] [addplot_option](#).

Y axis, X axis, Titles, Legend, Overall

`twoway_options` are any of the options documented in [G-3] [title_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

`teffects overlap` plots the estimated densities of the probability of getting each treatment level after `teffects`.

These plots can be used to check whether the overlap assumption is violated. The overlap assumption is satisfied when there is a chance of seeing observations in both the control and the treatment groups at each combination of covariate values; see [TE] [teffects intro](#) or [TE] [teffects intro advanced](#).

The overlap assumption is required by the estimators implemented in `teffects`. Intuitively, when the overlap assumption is violated, we cannot predict, or otherwise account for, the unobserved outcomes for some individuals.

There is evidence that the overlap assumption is violated when an estimated density has too much mass around 0 or 1; see [Busso, DiNardo, and McCrary \(2011\)](#). An implication of this point is that when the overlap assumption is violated, the estimated densities will have relatively little mass in the regions in which they overlap.

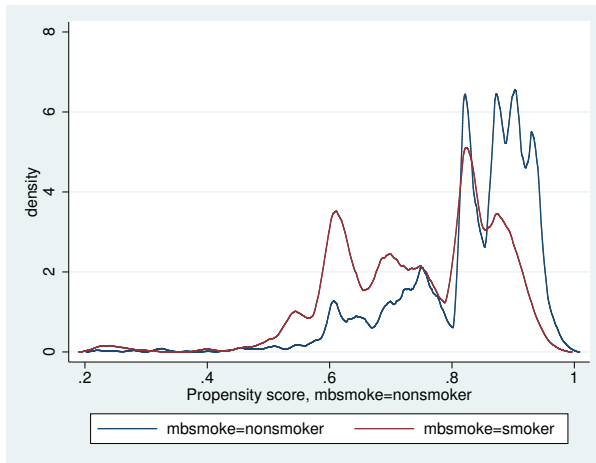
► Example 1: Assumption not violated

Continuing with [example 1](#) of [TE] [teffects ipw](#), we estimate the average treatment effect of smoking on birthweight and then draw the overlap plot:

```
. use http://www.stata-press.com/data/r13/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ipw (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu, probit)
Iteration 0:  EE criterion = 4.622e-21
Iteration 1:  EE criterion = 8.070e-26
Treatment-effects estimation          Number of obs      =      4642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: probit
```

	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
ATE						
mbsmoke (smoker vs nonsmoker)	-230.6886	25.81524	-8.94	0.000	-281.2856	-180.0917
P0mean						
mbsmoke nonsmoker	3403.463	9.571369	355.59	0.000	3384.703	3422.222


```
. teffects overlap
```



The graph displays the estimated density of the predicted probabilities that a nonsmoking mother is a nonsmoker and the estimated density of the predicted probabilities that a smoking mother is a nonsmoker.

Neither plot indicates too much probability mass near 0 or 1, and the two estimated densities have most of their respective masses in regions in which they overlap each other. Thus there is no evidence that the overlap assumption is violated.

◀

▶ Example 2: Assumption violated

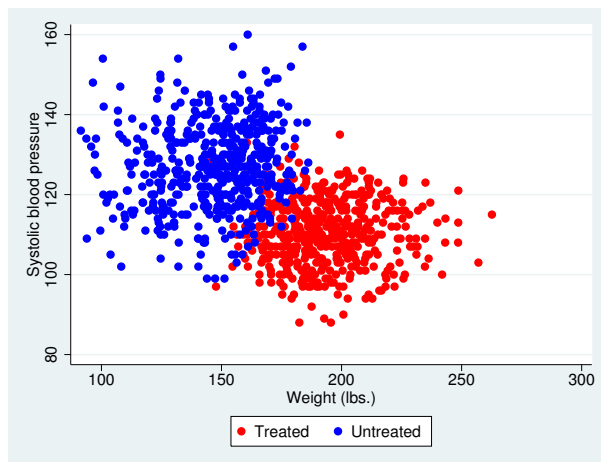
This example produces an overlap plot that indicates a failure of the overlap assumption. We will use simulated data, so we know that the assumption is not true.

In our simulated dataset, some of the 1,000 adult males were given drug XY1 for high blood pressure and others were not. A scatterplot of systolic blood pressure (`systolic`) and weight (`weight`) reveals that heavier men were given the treatment. (The scatterplots corresponding to the treatment group are colored red, while the scatterplots corresponding to the control group are colored blue.)

```

. use http://www.stata-press.com/data/r13/systolic2
(Systolic blood pressure)
. twoway (scatter systolic weight if xy1==1, mcolor(red))
>       (scatter systolic weight if xy1==0, mcolor(blue)),
>       legend(label(1 "Treated") label(2 "Untreated"))

```



There are no observations in the treated group for small weights, and there are no observations in the control group for large weights. There is clear evidence that the overlap assumption is violated.

Drawing an overlaid scatterplot is a straightforward way to check the overlap assumption in this example because there is only one covariate. This method is not available when there is more than one covariate. The predicted probability is a one-dimensional measure that captures the relevant multivariate information.

Below we estimate the parameters needed to calculate the predicted probabilities. The `pstolerance(1e-8)` option is specified to ensure that estimation is performed as long as the predicted probabilities are at least as large as $1e-8$.

```

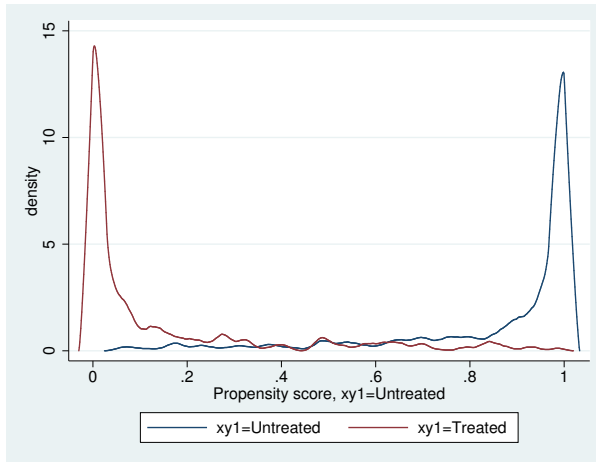
. teffects ipw (systolic) (xy1 weight), pstolerance(1e-8)
Iteration 0: EE criterion = 9.523e-18
Iteration 1: EE criterion = 8.779e-28
Treatment-effects estimation      Number of obs      =      1000
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: logit

```

systolic		Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
ATE	xy1 (Treated vs Untreated)	-16.23679	2.191703	-7.41	0.000	-20.53245	-11.94114
P0mean	xy1 Untreated	127.9094	.7004533	182.61	0.000	126.5365	129.2822

Now we can obtain the overlap plot.

```
. teffects overlap
```



The estimated density of the predicted probabilities that a treated individual is not assigned to XY1 treatment has most of its mass near 0. The estimated density of the predicted probabilities that an untreated individual is not assigned to XY1 treatment has most of its mass near 1. Note that the two have very little mass in the region in which they overlap. There is clear evidence that the overlap assumption is violated.

◀

Stored results

`teffects overlap` stores the following in `r()`:

Scalars

<code>r(bwidthj)</code>	kernel bandwidth for treatment level j
<code>r(nj)</code>	number of points at which the estimate was evaluated for treatment level j
<code>r(scalej)</code>	density bin width for treatment level j

Macros

<code>r(kernel)</code>	name of kernel
------------------------	----------------

References

- Busso, M., J. DiNardo, and J. McCrary. 2011. New evidence on the finite sample properties of propensity score reweighting and matching estimators. <http://emlab.berkeley.edu/~jmccrary/BDM2011.pdf>.
- Silverman, B. W. 1992. *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall.

Also see

[TE] **teffects** — Treatment-effects estimation for observational data

[TE] **teffects aipw** — Augmented inverse-probability weighting

[TE] **teffects ipw** — Inverse-probability weighting

[TE] **teffects ipwra** — Inverse-probability-weighted regression adjustment

[TE] **teffects nnmatch** — Nearest-neighbor matching

[TE] **teffects psmatch** — Propensity-score matching

[TE] **teffects ra** — Regression adjustment

Title

teffects postestimation — Postestimation tools for teffects

[Description](#)

[Syntax](#)

[Options](#)

[Remarks and examples](#)

[Also see](#)

Description

The following postestimation command is of special interest after `teffects`:

Command	Description
<code>teffects overlap</code>	overlap plots

The following standard postestimation commands are also available:

Command	Description
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estimates</code>	cataloging estimation results
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of coefficients
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
<code>predict</code>	predictions, residuals, influence statistics, and other diagnostic measures
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

Syntax

Syntaxes are presented under the following headings:

Syntax for predict after aipw and ipwra

Syntax for predict after ipw

Syntax for predict after nnmatch and psmatch

Syntax for predict after ra

Syntax for predict after aipw and ipwra

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

```
predict [type] { stub* | newvarlist } [if] [in], scores
```

<i>statistic</i>	Description
Main	
te	treatment effect; the default
cmean	conditional mean at treatment level
ps	propensity score
xb	linear prediction
psxb	linear prediction for propensity score
lnsigma	log square root of conditional latent variance (for outcome model <code>hetprobit()</code>) at treatment level
pslnsigma	log square root of latent variance (for treatment model <code>hetprobit()</code>) for propensity score

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

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

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

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

Syntax for predict after ipw

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

```
predict [type] { stub* | newvarlist } [if] [in], scores
```

<i>statistic</i>	Description
Main	
ps	propensity score; the default
xb	linear prediction for the propensity score
lnsigma	log square root of latent variance (for treatment model <code>hetprobit()</code>)

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` assumes `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`.

Syntax for predict after nnmatch and psmatch

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

<i>statistic</i>	Description
Main	
<code>te</code>	treatment effect; the default
<code>po</code>	potential outcome
<code>distance</code>	nearest-neighbor distance
<code>ps</code>	propensity score (<code>psmatch</code> only)
<code>lnsigma</code>	log square root of latent variance (for treatment model <code>hetprobit()</code>)

These statistics are available for the estimation sample only and require the estimation option `generate(stub)`. This is because of the nonparametric nature of the matching estimator.

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

You specify one new variable with `te` and `lnsigma`.

You specify one or two new variables with `po` and `ps`.

Syntax for predict after ra

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

```
predict [type] { stub* | newvarlist } [if] [in] [, scores]
```

<i>statistic</i>	Description
Main	
<code>te</code>	treatment effect; the default
<code>cmean</code>	conditional mean at treatment level
<code>xb</code>	linear prediction
<code>lnsigma</code>	log square root of conditional latent variance (for outcome model <code>hetprobit()</code>) 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 `lnsigma` assume `tlevel()` specifies the first treatment level.

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

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

Options

Options are presented under the following headings:

Options for predict after aipw and ipwra

Options for predict after ipw

Options for predict after nnmatch and psmatch

Options for predict after ra

Options for predict after aipw and 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.

xb calculates the 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.

psxb calculates the linear prediction for the propensity score 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).

lnsigma calculates the log square root of the conditional latent variance for each treatment level or the treatment level specified in `tlevel()`. This option is valid when outcome model `hetprobit()` was 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.

pslnsigma calculates the log square root of the latent variance for the propensity score. This option is only valid when treatment model `hetprobit()` was used. 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 propensity-score equations.

The j th new variable will contain the scores for the j th 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 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.

xb calculates the linear prediction for the propensity score 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).

lnsigma calculates the log square root of the latent variance. This option is only valid when treatment model `hetprobit()` was used. 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 propensity-score equations.

The j th new variable will contain the scores for the j th 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 nnmatch and psmatch

Main

`te`, the default, calculates the treatment effect.

`po` calculates the predicted potential outcomes for each observation and 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 new variables for the control and treated groups.

`distance` calculates the distances of the nearest neighbors for each observation. The number of variables generated is equal to the maximum number of nearest-neighbor matches. This is equal to the number of index variables generated by the estimation option `generate(stub)`. You may use the `stub*` syntax to set the distance variable prefix: `stub1, stub2, ...`

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

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

`tlevel(treat_level)` restricts potential-outcome estimation to either the treated group or the control group. This option may only be specified with options `po` and `ps`.

Options for predict after 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 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.

`lnsigma` calculates the log square root of the conditional latent variance for each treatment level or the treatment level specified in `tlevel()`. This option is valid when outcome model `hetprobit()` was 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 regression equations.

The j th new variable will contain the scores for the j th 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 `teffects`. `teffects overlap` provides a graphical method for checking the overlap assumption; see [TE] **teffects overlap**. 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 blow up.

We estimate the ATE of maternal smoking on infant birthweight by inverse-probability weighting; see [example 1](#) of [TE] **teffects ipw** for background.

```
. use http://www.stata-press.com/data/r13/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ipw (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu, probit)
Iteration 0:  EE criterion = 4.622e-21
Iteration 1:  EE criterion = 8.070e-26
Treatment-effects estimation      Number of obs      =      4642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: probit
```

	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
ATE						
mbsmoke (smoker vs nonsmoker)	-230.6886	25.81524	-8.94	0.000	-281.2856	-180.0917
POmean						
mbsmoke nonsmoker	3403.463	9.571369	355.59	0.000	3384.703	3422.222

Below we compute and summarize the estimated treatment probabilities.

```
. predict pr1
(option ps assumed; propensity score)
. summarize pr1 if mbsmoke==1, detail
      propensity score, mbsmoke=nonsmoker
```

Percentiles		Smallest		
1%	.2991634	.2196947		
5%	.544155	.2258079		
10%	.5973879	.2258079	Obs	864
25%	.63777	.2409025	Sum of Wgt.	864
50%	.7601717		Mean	.7456264
		Largest	Std. Dev.	.1276102
75%	.8453946	.9533503		
90%	.8943686	.9596144	Variance	.0162844
95%	.9096801	.961022	Skewness	-.7701643
99%	.9367017	.9665684	Kurtosis	3.858214

The smallest values do not imply very large weights.

Below we compute and summarize the estimated probabilities of not getting the treatment.

```
. generate pr0 = 1 -pr1
. summarize pr0 if mbsmoke==0, detail
      pr0
```

Percentiles		Smallest		
1%	.0351884	.0074551		
5%	.0578012	.0079309		
10%	.0674359	.0106305	Obs	3778
25%	.0950869	.0106305	Sum of Wgt.	3778
50%	.1372589		Mean	.1698913
		Largest	Std. Dev.	.1059434
75%	.2211142	.7547572		
90%	.3242757	.774192	Variance	.011224
95%	.3883457	.7803053	Skewness	1.514456
99%	.501537	.7816764	Kurtosis	6.151114

Although there are two small probabilities, overall the small values do not imply large weights.

Also see

[TE] [teffects overlap](#) — Overlap plots

[TE] [teffects aipw](#) — Augmented inverse-probability weighting

[TE] [teffects ipw](#) — Inverse-probability weighting

[TE] [teffects ipwra](#) — Inverse-probability-weighted regression adjustment

[TE] [teffects nnmatch](#) — Nearest-neighbor matching

[TE] [teffects psmatch](#) — Propensity-score matching

[TE] [teffects ra](#) — Regression adjustment

[U] [20 Estimation and postestimation commands](#)

Title

teffects psmatch — Propensity-score matching

Syntax

Remarks and examples

Also see

Menu

Stored results

Description

Methods and formulas

Options

References

Syntax

```
teffects psmatch (ovar) (tvar tmvarlist [, tmodel]) [if] [in] [weight]  
[ , stat options]
```

<i>tmodel</i>	Description
---------------	-------------

Model

logit	logistic treatment model; the default
probit	probit treatment model
hetprobit(<i>varlist</i>)	heteroskedastic probit treatment model

tmodel specifies the model for the treatment variable.

<i>stat</i>	Description
-------------	-------------

Stat

ate	estimate average treatment effect in population; the default
atet	estimate average treatment effect on the treated

<i>options</i>	Description
Model	
<code>nneighbor(#)</code>	specify number of matches per observation; default is <code>nneighbor(1)</code>
SE/Robust	
<code>vce(vcetype)</code>	<i>vcetype</i> may be <code>vce(robust [, nn(#)])</code> ; use robust Abadie–Imbens standard errors with # matches <code>vce(iid)</code> ; use default Abadie–Imbens standard errors
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>display_options</code>	control column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Advanced	
<code>caliper(#)</code>	specify the maximum distance for which two observations are potential neighbors
<code>pstolerance(#)</code>	set tolerance for in overlap assumption
<code>osample(newvar)</code>	<i>newvar</i> identifies observations that violate the overlap assumption
<code>control(# label)</code>	specify the level of <i>tvar</i> that is the control
<code>tlevel(# label)</code>	specify the level of <i>tvar</i> that is the treatment
<code>generate(stub)</code>	generate variables containing the observation numbers of the nearest neighbors
<code>coeflegend</code>	display legend instead of statistics

tvar must contain integer values representing the treatment levels.

tmvarlist may contain factor variables; see [U] 11.4.3 Factor variables.

`by` and `statsby` are allowed; see [U] 11.1.10 Prefix commands.

`fweights` are allowed; see [U] 11.1.6 weight.

`coeflegend` does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Menu

Statistics > Treatment effects > Matching estimators > Propensity-score matching

Description

`teffects psmatch` estimates treatment effects from observational data by propensity-score matching. PSM imputes the missing potential outcome for each subject by using an average of the outcomes of similar subjects that receive the other treatment level. Similarity between subjects is based on estimated treatment probabilities, known as propensity scores. The average treatment effect (ATE) is computed by taking the average of the difference between the observed and potential outcomes for each subject.

Options

Model

`nneighbor(#)` specifies the number of matches per observation. The default is `nneighbor(1)`. Each individual is matched with at least the specified number of individuals from the other treatment level. `nneighbor()` must specify an integer greater than or equal to 1 but no larger than the number of observations in the smallest group.

Stat

`stat` is one of two statistics: `ate` or `atet`. `ate` is the default.

`ate` specifies that the average treatment effect be estimated.

`atet` specifies that the average treatment effect on the treated be estimated.

SE/Robust

`vce(vcetype)` specifies the standard errors that are reported. By default, `teffects psmatch` uses two matches in estimating the robust standard errors.

`vce(robust [, nn(#)])` specifies that robust standard errors be reported and that the requested number of matches be used optionally.

`vce(iid)` specifies that standard errors for independently and identically distributed data be reported.

The standard derivative-based standard-error estimators cannot be used by `teffects psmatch`, because these matching estimators are not differentiable. The implemented method were derived by [Abadie and Imbens \(2006, 2011, 2012\)](#); see *Methods and formulas*.

As discussed in [Abadie and Imbens \(2008\)](#), bootstrap estimators do not provide reliable standard errors for the estimator implemented by `teffects psmatch`.

Reporting

`level(#)`; see [\[R\] estimation options](#).

`display_options`: `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `no!stretch`; see [\[R\] estimation options](#).

Advanced

`caliper(#)` specifies the maximum distance at which two observations are a potential match. By default, all observations are potential matches regardless of how dissimilar they are.

In `teffects psmatch`, the distance is measured by the estimated propensity score. If an observation has no matches, `teffects psmatch` exits with an error.

`pstolerance(#)` specifies the tolerance used to check the overlap assumption. The default value is `pstolerance(1e-5)`. `teffects` will exit with an error if an observation has an estimated propensity score smaller than that specified by `pstolerance()`.

`osample(newvar)` specifies that indicator variable `newvar` be created to identify observations that violate the overlap assumption. Two checks are made to verify the assumption. The first ensures that the propensity scores are greater than `pstolerance(#)` and less than $1 - \text{pstolerance}(\#)$. The second ensures that each observation has at least `nneighbor(#)` matches in the opposite treatment group within the distance specified by `caliper(#)`.

The `vce(robust, nn(#))` option also requires at least `#` matches in the same treatment group within the distance specified by `caliper(#)`.

The average treatment effect on the treated, option `atet`, using `vce(iid)` requires only `nneighbor(#)` control group matches for the treated group.

`control(# | label)` specifies the level of `tvar` that is the control. The default is the first treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(# | label)` specifies the level of `tvar` that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

`generate(stub)` specifies that the observation numbers of the nearest neighbors be stored in the new variables `stub1`, `stub2`, `...`. This option is required if you wish to perform postestimation based on the matching results. The number of variables generated may be more than `nneighbors(#)` because of tied distances. These variables may not already exist.

The following option is available with `teffects psmatch` but is not shown in the dialog box:

`coeflegend`; see [R] [estimation options](#).

Remarks and examples

Propensity-score matching uses an average of the outcomes of similar subjects who get the other treatment level to impute the missing potential outcome for each subject. The average treatment effect (ATE) is computed by taking the average of the difference between the observed and potential outcomes for each subject. `teffects psmatch` determines how near subjects are to each other by using estimated treatment probabilities, known as propensity scores. This type of matching is known as propensity-score matching (PSM).

PSM does not need bias correction, because PSM matches on a single continuous covariate. In contrast, the nearest-neighbor matching estimator implemented in `teffects nnmatch` uses a bias-correction term when matching on more than one continuous covariate. In effect, the PSM estimator parameterizes the bias-correction term in the treatment probability model. See [TE] [teffects intro](#) or [TE] [teffects intro advanced](#) for more information about this estimator.

We will illustrate the use of `teffects psmatch` by using data from a study of the effect of a mother's smoking status during pregnancy (`mbsmoke`) on infant birthweight (`bweight`) as reported by Cattaneo (2010). This dataset also contains information about each mother's age (`mage`), education level (`medu`), marital status (`mmarried`), whether the first prenatal exam occurred in the first trimester (`prenatal1`), whether this baby was the mother's first birth (`fbaby`), and the father's age (`fage`).

► Example 1: Estimating the ATE

We begin by using `teffects psmatch` to estimate the average treatment effect of `mbsmoke` on `bweight`. We use a logistic model (the default) to predict each subject's propensity score, using covariates `mage`, `medu`, `mmarried`, and `fbaby`. Because the performance of PSM hinges upon how well we can predict the propensity scores, we will use factor-variable notation to include both linear and quadratic terms for `mage`, the only continuous variable in our model:

```
. use http://www.stata-press.com/data/r13/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects psmatch (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu)
Treatment-effects estimation      Number of obs      =      4642
Estimator      : propensity-score matching      Matches: requested =      1
Outcome model  : matching                      min =          1
Treatment model: logit                      max =          74
```

	bweight	Coef.	AI Robust Std. Err.	z	P> z	[95% Conf. Interval]
ATE	mbsmoke (smoker vs nonsmoker)	-210.9683	32.021	-6.59	0.000	-273.7284 -148.2083

Smoking causes infants' birthweights to be reduced by an average of 211 grams.

◀

By default, `teffects psmatch` estimates the ATE by matching each subject to a single subject with the opposite treatment whose propensity score is closest. Sometimes, however, we may want to ensure that matching occurs only when the propensity scores of a subject and a match differ by less than a specified amount. To do that, we use the `caliper()` option. If a match within the distance specified in the `caliper()` option cannot be found, `teffects psmatch` exits.

▶ Example 2: Specifying the caliper

Here we reconsider the previous example, first specifying that we only want to consider a pair of observations a match if the absolute difference in the propensity scores is less than 0.03:

```
. teffects psmatch (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu),
> caliper(0.03)
no nearest-neighbor matches for observation 2209 within caliper 0.03; this
is not allowed
r(459);
```

The error arose because there is not a smoking mother whose propensity score is within 0.03 of the propensity score of the nonsmoking mother in observation 2209. If we instead raise the caliper to 0.10, we have matches for all subjects and therefore obtain the same results as in [example 1](#):


```
. teffects psmatch (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu),
> caliper(0.1)
Treatment-effects estimation      Number of obs      =      4642
Estimator      : propensity-score matching      Matches: requested =      1
Outcome model  : matching                      min =      1
Treatment model: logit                        max =      74
```

	bweight	Coef.	AI Robust Std. Err.	z	P> z	[95% Conf. Interval]
ATE	mbsmoke (smoker vs nonsmoker)	-210.9683	32.021	-6.59	0.000	-273.7284 -148.2083

◀

□ Technical note

[Example 2](#) highlights that estimating the ATE requires finding matches for both the treated and control subjects. In contrast, estimating the ATET only requires finding matches for the treated subjects. Because subject 2209 is a control subject, we can estimate the ATET using `caliper(0.03)`. We must also specify `vce(iid)` because the default robust standard errors for the estimated ATET require viable matches for both treated subjects and control subjects. (This requirement comes from the nonparametric method derived by [Abadie and Imbens \[2012\]](#).)

```
. teffects psmatch (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu),
> atet vce(iid) caliper(0.03)
Treatment-effects estimation      Number of obs      =      4642
Estimator      : propensity-score matching      Matches: requested =      1
Outcome model  : matching                      min =      1
Treatment model: logit                        max =      74
```

	bweight	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
ATET	mbsmoke (smoker vs nonsmoker)	-236.7848	26.11698	-9.07	0.000	-287.9731 -185.5964

□

In the previous examples, each subject was matched to at least one other subject, which is the default behavior for `teffects psmatch`. However, we can request that `teffects psmatch` match each subject to multiple subjects with the opposite treatment level by specifying the `nneighbor()` option. Matching on more distant neighbors can reduce the variance of the estimator at a cost of an increase in bias.

▷ Example 3

Now we request that `teffects psmatch` match a mother to four mothers in the opposite treatment group:

```
. teffects psmatch (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu),
> nneighbor(4)
Treatment-effects estimation           Number of obs       =    4642
Estimator       : propensity-score matching   Matches: requested =     4
Outcome model   : matching                   min =                4
Treatment model: logit                       max =                74
```

	Coef.	AI Robust Std. Err.	z	P> z	[95% Conf. Interval]	
<hr/>						
ATE						
mbsmoke (smoker vs nonsmoker)	-224.006	29.88627	-7.50	0.000	-282.582	-165.43

◀

Stored results

`teffects psmatch` stores the following in `e()`:

Scalars

```
e(N)                number of observations
e(nj)               number of observations for treatment level j
e(k_levels)         number of levels in treatment variable
e(caliper)          maximum distance between matches
e(treated)          level of treatment variable defined as treated
e(control)          level of treatment variable defined as control
e(k_nneighbor)      requested number of matches
e(k_nnmin)          minimum number of matches
e(k_nnmax)          maximum number of matches
e(k_robust)         matches for robust VCE
```

Macros

```
e(cmd)              teffects
e(cmdline)          command as typed
e(depvar)           name of outcome variable
e(tvar)             name of treatment variable
e(subcmd)           psmatch
e(tmodel)           logit, probit, or hetprobit
e(stat)             statistic estimated, ate or atet
e(wtype)           weight type
e(wexp)            weight expression
e(title)           title in estimation output
e(tlevels)         levels of treatment variable
e(psvarlist)       variables in propensity-score model
e(hvarlist)        variables for variance, only if hetprobit
e(vce)            vcetype specified in vce()
e(vcetype)         title used to label Std. Err.
e(datasignature)   the checksum
e(datasignaturevars) variables used in calculation of checksum
e(properties)      b V
e(estat_cmd)       program used to implement estat
e(predict)         program used to implement predict
e(marginsnotok)   predictions disallowed by margins
```

Matrices	
e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators
e(bps)	coefficient vector from propensity-score model
e(Vps)	variance–covariance matrix of the estimators from propensity-score model
Functions	
e(sample)	marks estimation sample

Methods and formulas

The methods and formulas used by `teffects psmatch` are documented in the *Methods and formulas* of [TE] `teffects nnmatch`.

References

- Abadie, A., and G. W. Imbens. 2006. Large sample properties of matching estimators for average treatment effects. *Econometrica* 74: 235–267.
- . 2008. On the failure of the bootstrap for matching estimators. *Econometrica* 76: 1537–1557.
- . 2011. Bias-corrected matching estimators for average treatment effects. *Journal of Business and Economic Statistics* 29: 1–11.
- . 2012. Matching on the estimated propensity score. Harvard University and National Bureau of Economic Research. <http://www.hks.harvard.edu/fs/aabadie/pscore.pdf>.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154.

Also see

- [TE] `teffects postestimation` — Postestimation tools for teffects
- [TE] `teffects` — Treatment-effects estimation for observational data
- [TE] `teffects nnmatch` — Nearest-neighbor matching
- [R] `hetprobit` — Heteroskedastic probit model
- [R] `logit` — Logistic regression, reporting coefficients
- [R] `mlogit` — Multinomial (polytomous) logistic regression
- [R] `probit` — Probit regression
- [U] **20 Estimation and postestimation commands**

Title

teffects ra — Regression adjustment

[Syntax](#)

[Remarks and examples](#)

[Also see](#)

[Menu](#)

[Stored results](#)

[Description](#)

[Methods and formulas](#)

[Options](#)

[References](#)

Syntax

```
teffects ra (ovar omvarlist [, omodel noconstant]) (tvar) [if] [in] [weight]  
[ , stat options]
```

<i>omodel</i>	Description
---------------	-------------

<code>linear</code>	linear outcome model; the default
<code>logit</code>	logistic outcome model
<code>probit</code>	probit outcome model
<code>hetprobit(<i>varlist</i>)</code>	heteroskedastic probit outcome model
<code>poisson</code>	exponential outcome model

omodel specifies the model for the outcome variable.

<i>stat</i>	Description
-------------	-------------

Stat

<code>ate</code>	estimate average treatment effect in population; the default
<code>atet</code>	estimate average treatment effect on the treated
<code>pomeans</code>	estimate potential-outcome means

<i>options</i>	Description
----------------	-------------

SE/Robust

<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>robust</code> , <code>bootstrap</code> , or <code>jackknife</code>
----------------------------------	--

Reporting

<code>llevel(#)</code>	set confidence level; default is <code>llevel(95)</code>
<code>aequations</code>	display auxiliary-equation results
<code>display_options</code>	control column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

Maximization

<code>maximize_options</code>	control the maximization process; seldom used
-------------------------------	---

Advanced

<code>control(# <i>label</i>)</code>	specify the level of <i>tvar</i> that is the control
<code>tlevel(# <i>label</i>)</code>	specify the level of <i>tvar</i> that is the treatment
<code>coeflegend</code>	display legend instead of statistics

tvar must contain integer values representing the treatment levels.

omvarlist may contain factor variables; see [U] 11.4.3 [Factor variables](#).

bootstrap, *by*, *jackknife*, and *statsby* are allowed; see [U] 11.1.10 [Prefix commands](#).

fweights and *iweights* are allowed; see [U] 11.1.6 [weight](#).

coeflegend does not appear in the dialog box.

See [U] 20 [Estimation and postestimation commands](#) for more capabilities of estimation commands.

Menu

Statistics > Treatment effects > Regression adjustment

Description

`teffects ra` estimates treatment effects from observational data via regression adjustment (RA). RA uses contrasts of averages of treatment-specific predicted outcomes to estimate treatment effects. `teffects ra` offers the choice of several different functional forms to facilitate continuous, binary, count, and nonnegative outcomes.

Options

Model

`noconstant`; see [R] [estimation options](#).

Stat

stat is one of three statistics: `ate`, `atet`, or `pomeans`. `ate` is the default.

`ate` specifies that the average treatment effect be estimated.

`atet` specifies that the average treatment effect on the treated be estimated.

`pomeans` specifies that the potential-outcome means for each treatment level be estimated.

SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`) and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce_option](#).

Reporting

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

`aequations` specifies that the results for the outcome-model or the treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

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

Maximization

`maximize_options`: `iterate(#)`, `[no]log`, and `from(init_specs)`; see [R] [maximize](#). These options are seldom used.

`init_specs` is one of

```
matname [ , skip copy ]
# [ , # ... ] , copy
```

Advanced

`control(# | label)` specifies the level of `tvar` that is the control. The default is the first treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with statistic `pomeans`. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(# | label)` specifies the level of `tvar` that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

The following option is available with `teffects ra` but is not shown in the dialog box:

`coeflegend`; see [R] [estimation options](#).

Remarks and examples

Remarks are presented under the following headings:

[Overview](#)

[Video example](#)

Overview

Regression adjustment (RA) estimators use the contrasts of the averages of treatment-specific predicted outcomes to estimate treatment effects. RA estimators use a two-step approach to estimating treatment effects:

1. They fit separate regression models of the outcome on a set of covariates for each treatment level.
2. They compute the averages of the predicted outcomes for each subject and treatment level. These averages reflect the POMS. The contrasts of these averages provide estimates of the ATES. By restricting the computations of the means to the subset of treated subjects, we obtain the ATETs.

RA estimators are consistent as long as the treatment is independent of the potential outcomes after conditioning on the covariates. In fact, `teffects ra` uses an estimation technique that implements both steps at once so that we do not need to correct the standard errors in the second step to reflect the uncertainty surrounding the predicted outcomes.

We will illustrate the use of `teffects ra` by using data from a study of the effect of a mother's smoking status during pregnancy (`mbsmoke`) on infant birthweight (`bweight`) as reported by [Cattaneo \(2010\)](#). This dataset also contains information about each mother's age (`mage`), education level (`medu`), marital status (`mmarried`), whether the first prenatal exam occurred in the first trimester (`prenatal1`), and whether this baby was the mother's first birth (`fbaby`).

▷ Example 1: Estimating the ATE

We begin by using `teffects ra` to estimate the average treatment effect of smoking, controlling for first-trimester exam status, marital status, mother's age, and first-birth status. In Stata, we type

```
. use http://www.stata-press.com/data/r13/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ra (bweight prenatal1 mmarried mage fbaby) (mbsmoke)
Iteration 0: EE criterion = 5.288e-25
Iteration 1: EE criterion = 4.495e-26
Treatment-effects estimation          Number of obs    =    4642
Estimator      : regression adjustment
Outcome model  : linear
Treatment model: none
```

bweight	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
ATE						
mbsmoke (smoker vs nonsmoker)	-239.6392	23.82402	-10.06	0.000	-286.3334	-192.945
P0mean						
mbsmoke nonsmoker	3403.242	9.525207	357.29	0.000	3384.573	3421.911

Smoking causes infants' birthweights to be reduced by an average of 240 grams from the average of 3,403 grams for infants of mothers who do not smoke.

◀

The previous results showed us the average amount by which infants' weights are affected by their mothers' decision to smoke. We may instead be interested in knowing the average amount by which the weight of babies born to smoking mothers was decreased as a result of smoking. The `ATE` provides us with the answer.

▷ Example 2: Estimating the ATET

To obtain the ATET rather than the ATE, we use the `atet` option:

```
. teffects ra (bweight prenatal1 mmarried mage fbaby) (mbsmoke), atet
Iteration 0: EE criterion = 4.528e-25
Iteration 1: EE criterion = 4.047e-26

Treatment-effects estimation          Number of obs      =      4642
Estimator       : regression adjustment
Outcome model   : linear
Treatment model : none
```

bweight	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
ATET						
mbsmoke (smoker vs nonsmoker)	-223.3017	22.7422	-9.82	0.000	-267.8755	-178.7278
POmean						
mbsmoke nonsmoker	3360.961	12.75749	263.45	0.000	3335.957	3385.966

The ATET differs from the ATE because the distribution of the covariates among mothers who smoke differs from the distribution for nonsmoking mothers. For example, in [TE] **teffects intro**, we remarked that in our sample, mothers who smoked tended to be older than those who did not. The differing distributions of covariates also affect the estimated POMs.

◀

By default, `teffects ra` reports the ATE, which is the difference between the two POMs in the case of a binary treatment variable. Sometimes, we want to know the estimated POMs themselves. We might also want to see the actual regression equations used to estimate the POMs. Obtaining this information is easy, as the next example illustrates.

▷ Example 3: Estimating the POMs

Here we use the `pomeans` option to display the POMs and the `aequations` option to display the estimated regression coefficients for the treated and untreated subjects.

```
. teffects ra (bweight prenatal1 mmarried mage fbaby) (mbsmoke),
> pomeans aequations
Iteration 0:   EE criterion = 5.288e-25
Iteration 1:   EE criterion = 5.204e-26
Treatment-effects estimation          Number of obs   =       4642
Estimator      : regression adjustment
Outcome model  : linear
Treatment model: none
```

bweight	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
POMeans						
mbsmoke						
nonsmoker	3403.242	9.525207	357.29	0.000	3384.573	3421.911
smoker	3163.603	21.86351	144.70	0.000	3120.751	3206.455
OME0						
prenatal1	64.40859	27.52699	2.34	0.019	10.45669	118.3605
mmarried	160.9513	26.6162	6.05	0.000	108.7845	213.1181
mage	2.546828	2.084324	1.22	0.222	-1.538373	6.632028
fbaby	-71.3286	19.64701	-3.63	0.000	-109.836	-32.82117
_cons	3202.746	54.01082	59.30	0.000	3096.886	3308.605
OME1						
prenatal1	25.11133	40.37541	0.62	0.534	-54.02302	104.2457
mmarried	133.6617	40.86443	3.27	0.001	53.5689	213.7545
mage	-7.370881	4.21817	-1.75	0.081	-15.63834	.8965804
fbaby	41.43991	39.70712	1.04	0.297	-36.38461	119.2644
_cons	3227.169	104.4059	30.91	0.000	3022.537	3431.801

The nonsmoker POM for infant birthweight is 3,403 grams; that means that if none of the women in our sample smoked during pregnancy, the expected average birthweight would be 3,403 grams. The POM if all mothers did smoke during pregnancy is 3,164 grams, a difference of 240 grams, as we established in [example 1](#). The coefficients for the equation labeled OME0 represent the linear equation used to estimate the nontreated POM, and the coefficients for the equation labeled OME1 represent the linear equation used to estimate the treated POM. The coefficients are identical to those we would obtain using `regress`, but the standard errors differ slightly because `teffects ra` does not make the small-sample adjustment that `regress` does.

◀

We often express statistics as percentages to alleviate scaling issues and aid interpretation. In the present context, we may wish to express an ATE as a percentage of the untreated POM to gain a more intuitive measure of efficacy.

▷ Example 4: Reporting the ATE as a percentage

Sometimes, we are interested in reporting the estimated treatment effect as a percentage of the untreated POM. We continue to use the same model as in the previous examples, but we specify the `coeflegend` option so that `teffects ra` reports the names of the parameters. Knowing the correct names to use, we can then use `nlcom` to obtain the percentage change along with its delta-method-based standard error. We type

```

. use http://www.stata-press.com/data/r13/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ra (bweight prenatal1 mmarried mage fbaby) (mbsmoke), coeflegend
Iteration 0:  EE criterion = 5.288e-25
Iteration 1:  EE criterion = 4.495e-26
Treatment-effects estimation          Number of obs   =   4642
Estimator      : regression adjustment
Outcome model  : linear
Treatment model: none

```

bweight	Coef.	Legend
ATE		
mbsmoke (smoker vs nonsmoker)	-239.6392	_b[ATE:r1vs0.mbsmoke]
POmean		
mbsmoke nonsmoker	3403.242	_b[POmean:r0.mbsmoke]

```

. nlcom _b[ATE:r1vs0.mbsmoke] / _b[POmean:r0.mbsmoke]
       _nl_1:  _b[ATE:r1vs0.mbsmoke] / _b[POmean:r0.mbsmoke]

```

bweight	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
_nl_1	-.070415	.0069245	-10.17	0.000	-.0839867 -0.0568433

We find that smoking during pregnancy reduces birthweight by about 7% on average, a statistically significant amount.

◀

Birthweights cannot be negative, though it is possible for a linear regression model to make negative predictions. A common way to enforce nonnegative predictions is to use an exponential conditional-mean model, which is commonly fitted using the Poisson quasimaximum-likelihood estimator, as discussed in [Cameron and Trivedi \(2005, sec. 5.7\)](#), [Wooldridge \(2010, sec. 18.2\)](#), and [Pawitan \(2001, chap. 14\)](#). `teffects ra` provides an option to use this model rather than linear regression for the outcomes.

▶ Example 5: Modeling nonnegative outcomes

Now we refit our model of smoking behavior on birthweight, but we specify the `poisson` option in the outcome-model equation so that `teffects ra` uses the Poisson exponential model rather than linear regression:

```

. teffects ra (bweight prenatal1 mmarried mage fbaby, poisson) (mbsmoke)
Iteration 0:   EE criterion = 3.949e-17
Iteration 1:   EE criterion = 1.246e-23
Treatment-effects estimation      Number of obs      =      4642
Estimator      : regression adjustment
Outcome model  : Poisson
Treatment model: none

```

bweight	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
ATE						
mbsmoke (smoker vs nonsmoker)	-239.6669	23.83757	-10.05	0.000	-286.3877	-192.9462
POmean						
mbsmoke nonsmoker	3403.178	9.526006	357.25	0.000	3384.508	3421.849

In this case, using a model that forces outcomes to be nonnegative did not make any substantive difference. In this dataset, nearly 90% of babies weigh at least 2,700 grams, and even the smallest baby weighs 340 grams. When the dependent variable is so large, the predictions from Poisson and linear regression models are remarkably similar.

◀

Video example

[Treatment effects: Regression adjustment](#)

Stored results

`teffects ra` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level j
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	<code>teffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(subcmd)</code>	<code>ra</code>
<code>e(omodel)</code>	linear, logit, probit, hetprobit, or poisson
<code>e(stat)</code>	statistic estimated, ate, atet, or pomeans
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. Err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

Methods and formulas

`teffects ra` implements a smooth treatment-effects estimator. All smooth treatment-effects estimators are documented in [TE] [teffects aipw](#).

References

- 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.
- Pawitan, Y. 2001. *In All Likelihood: Statistical Modelling and Inference Using Likelihood*. Oxford: Oxford University Press.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

- [TE] [teffects postestimation](#) — Postestimation tools for teffects
- [TE] [teffects](#) — Treatment-effects estimation for observational data
- [R] [hetprobit](#) — Heteroskedastic probit model
- [R] [logit](#) — Logistic regression, reporting coefficients
- [R] [poisson](#) — Poisson regression
- [R] [probit](#) — Probit regression
- [R] [regress](#) — Linear regression
- [U] [20 Estimation and postestimation commands](#)

Glossary

AIPW estimator. See *augmented inverse-probability-weighted estimator*.

ATE. See *average treatment effect*.

ATET. See *average treatment effect on the treated*.

augmented inverse-probability-weighted estimator. An augmented inverse-probability-weighted (AIPW) estimator is an inverse-probability-weighted estimator that includes an augmentation term that corrects the estimator when the treatment model is misspecified. When the treatment is correctly specified, the augmentation term vanishes as the sample size becomes large. An AIPW estimator uses both an outcome model and a treatment model and is a doubly robust estimator.

average treatment effect. The average treatment effect is the average effect of the treatment among all individuals in a population.

average treatment effect on the treated. The average treatment effect on the treated is the average effect of the treatment among those individuals who actually get the treatment.

CI assumption. See *conditional-independence assumption*.

conditional mean. The conditional mean expresses the average of one variable as a function of some other variables. More formally, the mean of y conditional on \mathbf{x} is the mean of y for given values of \mathbf{x} ; in other words, it is $E(y|\mathbf{x})$.

A conditional mean is also known as a regression or as a conditional expectation.

conditional-independence assumption. The conditional-independence assumption requires that the common variables that affect treatment assignment and treatment-specific outcomes be observable. The dependence between treatment assignment and treatment-specific outcomes can be removed by conditioning on these observable variables.

This assumption is also known as a selection-on-observables assumption because its central tenet is the observability of the common variables that generate the dependence.

counterfactual. A counterfactual is an outcome a subject would have obtained had that subject received a different level of treatment. In the binary-treatment case, the counterfactual outcome for a person who received treatment is the outcome that person would have obtained had the person instead not received treatment; similarly, the counterfactual outcome for a person who did not receive treatment is the outcome that person would have obtained had the person received treatment.

Also see *potential outcome*.

doubly robust estimator. A doubly robust estimator only needs one of two auxiliary models to be correctly specified to estimate a parameter of interest.

Doubly robust estimators for treatment effects are consistent when either the outcome model or the treatment model is correctly specified.

EE estimator. See *estimating-equation estimator*.

estimating-equation estimator. An estimating-equation (EE) estimator calculates parameter estimates by solving a system of equations. Each equation in this system is the sample average of a function that has mean zero.

These estimators are also known as M estimators or Z estimators in the statistics literature and as generalized method of moments (GMM) estimators in the econometrics literature.

i.i.d. sampling assumption. See *independent and identically distributed sampling assumption*.

independent and identically distributed sampling assumption. The independent and identically distributed (i.i.d.) sampling assumption specifies that each observation is unrelated to (independent of) all the other observations and that each observation is a draw from the same (identical) distribution.

individual-level treatment effect. An individual-level treatment effect is the difference in an individual's outcome that would occur because this individual is given one treatment instead of another. In other words, an individual-level treatment effect is the difference between two potential outcomes for an individual.

For example, the blood pressure an individual would obtain after taking a pill minus the blood pressure an individual would obtain had that person not taken the pill is the individual-level treatment effect of the pill on blood pressure.

inverse-probability-weighted estimators. Inverse-probability-weighted (IPW) estimators use weighted averages of the observed outcome variable to estimate the potential-outcome means. The weights are the reciprocals of the treatment probabilities estimated by a treatment model.

inverse-probability-weighted regression-adjustment estimators.

Inverse-probability-weighted regression-adjustment (IPWRA) estimators use the reciprocals of the estimated treatment probability as weights to estimate missing-data-corrected regression coefficients that are subsequently used to compute the potential-outcome means.

IPW estimators. See *inverse-probability-weighted estimators*.

IPWRA estimators. See *inverse-probability-weighted regression-adjustment estimators*.

matching estimator. An estimator that compares differences between the outcomes of similar—that is, matched—individuals. Each individual that receives a treatment is matched to a similar individual that does not get the treatment, and the difference in their outcomes is used to estimate the individual-level treatment effect. Likewise, each individual that does not receive a treatment is matched to a similar individual that does get the treatment, and the difference in their outcomes is used to estimate the individual-level treatment effect.

multivalued treatment effect. A multivalued treatment refers to a treatment that has more than two values. For example, a person could have taken a 20 mg dose of a drug, a 40 mg dose of the drug, or not taken the drug at all.

nearest-neighbor matching. Nearest-neighbor matching uses the distance between observed variables to find similar individuals.

observational data. In observational data, treatment assignment is not controlled by those who collected the data; thus some common variables affect treatment assignment and treatment-specific outcomes.

outcome model. An outcome model is a model used to predict the outcome as a function of covariates and parameters.

overlap assumption. The overlap assumption requires that each individual have a positive probability of each possible treatment level.

POMs. See *potential-outcome means*.

potential outcome. The potential outcome is the outcome an individual would obtain if given a specific treatment.

For example, an individual has one potential blood pressure after taking a pill and another potential blood pressure had that person not taken the pill.

potential-outcome means. The potential-outcome means refers to the means of the potential outcomes for a specific treatment level.

The mean blood pressure if everyone takes a pill and the mean blood pressure if no one takes a pill are two examples.

The average treatment effect is the difference between potential-outcome mean for the treated and the potential-outcome mean for the not treated.

propensity score. The propensity score is the probability that an individual receives a treatment.

propensity-score matching. Propensity-score matching uses the distance between estimated propensity scores to find similar individuals.

regression-adjustment estimators. Regression-adjustment estimators use means of predicted outcomes for each treatment level to estimate each potential-outcome mean.

selection-on-observables. See *conditional-independence assumption*.

smooth treatment-effects estimator. A smooth treatment-effects estimator is a smooth function of the data so that standard methods approximate the distribution of the estimator. The RA, IPW, AIPW, and IPWRA estimators are all smooth treatment-effects estimators while the nearest-neighbor matching estimator and the propensity-score matching estimator are not.

treatment model. A treatment model is a model used to predict treatment-assignment probabilities as a function of covariates and parameters.

unconfoundedness. See *conditional-independence assumption*.

Subject and author index

This is the subject and author index for the *Treatment-Effects Reference Manual*. Readers interested in topics other than treatment effects should see the combined [subject index](#) (and the [combined author index](#)) in the *Glossary and Index*.

A

Abadie, A., [TE] [teffects intro advanced](#), [TE] [teffects multivalued](#), [TE] [teffects nnmatch](#), [TE] [teffects psmatch](#)

Abadie–Imbens robust standard errors, see [robust](#), [Abadie–Imbens standard errors](#)

AIPW, see [augmented inverse-probability weighting](#) [aipw](#), [teffects](#) subcommand, [TE] [teffects aipw](#)

Angrist, J. D., [TE] [teffects intro advanced](#)

ATE, see [average treatment effect](#)

ATE and ATET, comparing, [TE] [teffects intro advanced](#), *also see* [average treatment effect](#), *also see* [average treatment effect on treated](#)

ATET, see [average treatment effect on treated](#)

augmented inverse-probability weighting, [TE] [teffects aipw](#), [TE] [teffects intro](#), [TE] [teffects intro advanced](#), [TE] [Glossary](#)

average treatment effect, [TE] [teffects aipw](#), [TE] [teffects intro](#), [TE] [teffects intro advanced](#), [TE] [teffects ipw](#), [TE] [teffects ipwra](#), [TE] [teffects multivalued](#), [TE] [teffects nnmatch](#), [TE] [teffects psmatch](#), [TE] [teffects ra](#), [TE] [Glossary](#)

average treatment effect on treated, [TE] [teffects intro](#), [TE] [teffects intro advanced](#), [TE] [teffects ipw](#), [TE] [teffects ipwra](#), [TE] [teffects multivalued](#), [TE] [teffects nnmatch](#), [TE] [teffects psmatch](#), [TE] [teffects ra](#), [TE] [Glossary](#)

B

Bang, H., [TE] [teffects intro advanced](#)

Barnow, B. S., [TE] [etregress](#)

Becker, S. O., [TE] [teffects intro advanced](#)

biweight kernel function, [TE] [teffects overlap](#)

Blundell, R., [TE] [teffects intro advanced](#), [TE] [teffects multivalued](#)

Boos, D. D., [TE] [teffects aipw](#)

Brave, S., [TE] [etregress](#)

Brown, G. K., [TE] [etregress](#), [TE] [teffects intro advanced](#)

Busso, M., [TE] [teffects overlap](#)

C

Cain, G. G., [TE] [etregress](#)

Caliendo, M., [TE] [teffects intro advanced](#)

Cameron, A. C., [TE] [etregress](#), [TE] [teffects aipw](#), [TE] [teffects intro advanced](#), [TE] [teffects ra](#)

Cattaneo, M. D., [TE] [teffects aipw](#), [TE] [teffects intro](#), [TE] [teffects intro advanced](#), [TE] [teffects ipw](#), [TE] [teffects ipwra](#), [TE] [teffects multivalued](#), [TE] [teffects nnmatch](#), [TE] [teffects psmatch](#), [TE] [teffects ra](#)

CI assumption, see [conditional-independence assumption](#)

cluster estimator of variance

treatment-effects model, [TE] [etpoisson](#), [TE] [etregress](#)

CMI assumption, see [conditional mean independence assumption](#)

conditional mean, [TE] [Glossary](#)

conditional mean independence assumption, [TE] [teffects intro advanced](#)

conditional-independence assumption, [TE] [teffects intro](#), [TE] [teffects intro advanced](#), [TE] [Glossary](#)

confidence interval for incidence-rate ratios, [TE] [etpoisson](#)

Cong, R., [TE] [etregress](#)

constrained estimation, treatment-effects model, [TE] [etpoisson](#), [TE] [etregress](#)

cosine kernel function, [TE] [teffects overlap](#)

counterfactual, [TE] [Glossary](#), *also see* [potential outcome](#)

Cousens, S. N., [TE] [teffects intro advanced](#)

D

Daniel, R. M., [TE] [teffects intro advanced](#)

data, observational, see [observational data](#)

De Stavola, B. L., [TE] [teffects intro advanced](#)

Dearden, L., [TE] [teffects intro advanced](#), [TE] [teffects multivalued](#)

DiNardo, J., [TE] [teffects overlap](#)

doubly robust estimator, [TE] [teffects aipw](#), [TE] [teffects intro](#), [TE] [teffects intro advanced](#), [TE] [teffects ipwra](#), [TE] [Glossary](#)

Drukker, D. M., [TE] [etregress](#), [TE] [teffects aipw](#), [TE] [teffects intro advanced](#), [TE] [teffects multivalued](#), [TE] [teffects nnmatch](#)

Dunn, G., [TE] [teffects multivalued](#)

E

EE estimator, see [estimating-equation estimator](#)

Emsley, R., [TE] [teffects multivalued](#)

endogenous treatment, [TE] [etpoisson](#), [TE] [etregress](#)

Epanechnikov kernel function, [TE] [teffects overlap](#)

estimating-equation estimator, [TE] [teffects aipw](#), [TE] [teffects ipw](#), [TE] [teffects ipwra](#), [TE] [teffects ra](#), [TE] [Glossary](#)

etpoisson command, [TE] [etpoisson](#), [TE] [etpoisson postestimation](#)

etregress command, [TE] [etregress](#), [TE] [etregress postestimation](#)

F

Frölich, M., [TE] [teffects multivalued](#)

G

Gaussian kernel function, [TE] [teffects overlap](#)

Glidden, D. V., [TE] [teffects intro advanced](#)

Goldberger, A. S., [TE] [etregress](#)

graphs,

density, [TE] [teffects overlap](#)

treatment-effects overlap, [TE] [teffects overlap](#)

Greene, W. H., [TE] [etpoisson](#), [TE] [etregress](#)

H

hazard, [TE] [etregress](#)

Heckman, J., [TE] [etregress](#), [TE] [teffects intro advanced](#)

Herr, J. L., [TE] [teffects intro advanced](#), [TE] [teffects nnmatch](#)

Hirano, K., [TE] [teffects intro advanced](#)

Holland, A. D., [TE] [teffects aipw](#), [TE] [teffects intro advanced](#), [TE] [teffects multivalued](#)

Holland, P. W., [TE] [teffects intro advanced](#)

Horvitz, D. G., [TE] [teffects intro advanced](#)

Huber/White/sandwich estimator of variance, see [robust](#), [Huber/White/sandwich estimator of variance](#)

I

i.i.d. assumption, see [independent and identically distributed sampling assumption](#)

Imbens, G. W., [TE] [teffects intro advanced](#), [TE] [teffects multivalued](#), [TE] [teffects nnmatch](#), [TE] [teffects psmatch](#)

incidence-rate ratio estimation, Poisson regression, [TE] [etpoisson](#)

independent and identically distributed sampling assumption, [TE] [teffects intro](#), [TE] [teffects intro advanced](#), [TE] [Glossary](#)

individual-level treatment effect, [TE] [Glossary](#)

inverse-probability weighting, [TE] [teffects intro](#), [TE] [teffects intro advanced](#), [TE] [teffects ipw](#), [TE] [Glossary](#)

inverse-probability-weighted regression adjustment, [TE] [teffects intro](#), [TE] [teffects intro advanced](#), [TE] [teffects ipwra](#), [TE] [Glossary](#)

IPW, see [inverse-probability weighting](#)

ipw, [teffects](#) subcommand, [TE] [teffects ipw](#)

IPWRA, see [inverse-probability-weighted regression adjustment](#)

ipwra, [teffects](#) subcommand, [TE] [teffects ipwra](#)

K

Kang, J. D. Y., [TE] [teffects intro advanced](#)

L

Lane, P. W., [TE] [teffects intro advanced](#)

Lei-Gomez, Q., [TE] [teffects intro advanced](#)

Leuven, E., [TE] [teffects intro advanced](#)

limited dependent variables, [TE] [etpoisson](#)

linear regression, [TE] [etregress](#), [TE] [teffects ra](#)

log-linear model, [TE] [etpoisson](#)

Lunt, M., [TE] [teffects multivalued](#)

M

Maddala, G. S., [TE] [etregress](#)

matching estimator, [TE] [teffects intro](#), [TE] [teffects intro advanced](#), [TE] [teffects nnmatch](#), [TE] [teffects psmatch](#), [TE] [Glossary](#)

McCrary, J., [TE] [teffects overlap](#)

McCulloch, C. E., [TE] [teffects intro advanced](#)

McFadden, D., [TE] [teffects aipw](#)

Melly, B., [TE] [teffects multivalued](#)

Mergoupis, T., [TE] [etregress](#), [TE] [teffects intro advanced](#)

multivalued treatment effect, [TE] [teffects aipw](#), [TE] [teffects ipw](#), [TE] [teffects ipwra](#), [TE] [teffects multivalued](#), [TE] [teffects ra](#), [TE] [Glossary](#)

N

Nannicini, T., [TE] [etregress](#)

Navarro-Lozano, S., [TE] [teffects intro advanced](#)

nearest-neighbor matching, [TE] [teffects intro](#), [TE] [teffects intro advanced](#), [TE] [teffects nnmatch](#), [TE] [Glossary](#)

Nelder, J. A., [TE] [teffects intro advanced](#)

Newey, W. K., [TE] [teffects aipw](#)

Nichols, A., [TE] [etregress](#), [TE] [teffects intro advanced](#)

nnmatch, [teffects](#) subcommand, [TE] [teffects nnmatch](#)

nonlinear regression, [TE] [teffects ra](#)

nonparametric analysis, treatment effect, [TE] [teffects nnmatch](#), [TE] [teffects psmatch](#)

O

observational data, [TE] [etpoisson](#), [TE] [etpoisson postestimation](#), [TE] [etregress](#), [TE] [etregress postestimation](#), [TE] [teffects](#), [TE] [teffects aipw](#), [TE] [teffects intro](#), [TE] [teffects intro advanced](#), [TE] [teffects ipw](#), [TE] [teffects ipwra](#), [TE] [teffects nnmatch](#), [TE] [teffects overlap](#), [TE] [teffects postestimation](#), [TE] [teffects psmatch](#), [TE] [teffects ra](#), [TE] [treatment effects](#), [TE] [Glossary](#)

outcome model, [TE] [etpoisson](#), [TE] [etregress](#), [TE] [teffects aipw](#), [TE] [teffects intro advanced](#), [TE] [teffects ipwra](#), [TE] [teffects ra](#), [TE] [Glossary](#)

outcomes,

count, Poisson, [TE] **etpoisson**

ordinal, treatment effect, [TE] **teffects multivalued**

overlap assumption, [TE] **teffects intro**, [TE] **teffects intro advanced**, [TE] **teffects overlap**, [TE] **Glossary**

overlap, **teffects** subcommand, [TE] **teffects overlap**

P

Parzen kernel function, [TE] **teffects overlap**

Pawitan, Y., [TE] **teffects ra**

Pickles, A., [TE] **teffects multivalued**

Pischke, J.-S., [TE] **teffects intro advanced**

POMs, see potential-outcome means

postestimation command, [TE] **teffects overlap**, [TE] **teffects postestimation**

potential outcome, [TE] **etpoisson**, [TE] **etpoisson postestimation**, [TE] **etregress**, [TE] **teffects aipw**, [TE] **teffects intro**, [TE] **teffects intro advanced**, [TE] **teffects ipw**, [TE] **teffects ipwra**, [TE] **teffects nnmatch**, [TE] **teffects overlap**, [TE] **teffects postestimation**, [TE] **teffects psmatch**, [TE] **teffects ra**, [TE] **Glossary**

potential-outcome means, [TE] **teffects aipw**, [TE] **teffects intro**, [TE] **teffects intro advanced**, [TE] **teffects ipw**, [TE] **teffects ipwra**, [TE] **teffects multivalued**, [TE] **teffects ra**, [TE] **Glossary**

predict command, [TE] **teffects postestimation**

propensity score, [TE] **teffects intro**, [TE] **teffects intro advanced**, [TE] **teffects postestimation**, [TE] **teffects psmatch**, [TE] **Glossary**

propensity-score matching, [TE] **teffects intro**, [TE] **teffects intro advanced**, [TE] **teffects psmatch**, [TE] **Glossary**

psmatch, **teffects** subcommand, [TE] **teffects psmatch**

R

ra, **teffects** subcommand, [TE] **teffects ra**

rectangle kernel function, [TE] **teffects overlap**

regression adjustment, [TE] **teffects intro**, [TE] **teffects intro advanced**, [TE] **teffects ra**, [TE] **Glossary**

Ridder, G., [TE] **teffects intro advanced**

Robins, J. M., [TE] **teffects intro advanced**

robust, Abadie–Imbens standard errors, [TE] **teffects nnmatch**, [TE] **teffects psmatch**

robust, Huber/White/sandwich estimator of variance, Poisson regression, treatment effect, [TE] **etpoisson** treatment effect, [TE] **etpoisson**, [TE] **etregress**, [TE] **teffects aipw**, [TE] **teffects ipw**, [TE] **teffects ipwra**, [TE] **teffects ra**

Rosenbaum, P. R., [TE] **teffects intro advanced**

Rotnitzky, A., [TE] **teffects intro advanced**

Rubin, D. B., [TE] **teffects intro advanced**

S

Sándor, L., [TE] **teffects intro advanced**

sandwich/Huber/White estimator of variance, see robust, Huber/White/sandwich estimator of variance

Schafer, J. L., [TE] **teffects intro advanced**

selection-on-observables, see conditional-independence assumption

Shiboski, S. C., [TE] **teffects intro advanced**

Sianesi, B., [TE] **teffects intro advanced**, [TE] **teffects multivalued**

Silverman, B. W., [TE] **teffects overlap**

smooth treatment-effects matching, [TE] **teffects aipw**, [TE] **teffects ipw**, [TE] **teffects ipwra**, [TE] **teffects ra**, [TE] **Glossary**

stable unit treatment value assumption, [TE] **teffects intro advanced**

standard errors, robust, see robust, Abadie–Imbens standard errors, see robust, Huber/White/sandwich estimator of variance

Stefanski, L. A., [TE] **teffects aipw**

Sued, M., [TE] **teffects intro advanced**

SUTVA, see stable unit treatment value assumption

T

Tan, Z., [TE] **teffects aipw**, [TE] **teffects intro advanced**

teffects, [TE] **teffects**, [TE] **teffects postestimation aipw** command, [TE] **teffects aipw ipw** command, [TE] **teffects ipw ipwra** command, [TE] **teffects ipwra nnmatch** command, [TE] **teffects nnmatch overlap** command, [TE] **teffects overlap psmatch** command, [TE] **teffects psmatch ra** command, [TE] **teffects ra**

Terza, J. V., [TE] **etpoisson**

Thompson, D. J., [TE] **teffects intro advanced**

treatment

effect, [TE] **etpoisson**, [TE] **etregress**, [TE] **teffects**, [TE] **teffects aipw**, [TE] **teffects intro**, [TE] **teffects intro advanced**, [TE] **teffects ipw**, [TE] **teffects ipwra**, [TE] **teffects multivalued**, [TE] **teffects nnmatch**, [TE] **teffects overlap**, [TE] **teffects postestimation**, [TE] **teffects psmatch**, [TE] **teffects ra**, [TE] **treatment effects**

model, [TE] **etpoisson**, [TE] **etregress**, [TE] **teffects aipw**, [TE] **teffects intro advanced**, [TE] **teffects ipw**, [TE] **teffects ipwra**, [TE] **teffects psmatch**, [TE] **Glossary**

triangle kernel function, [TE] **teffects overlap**

Trivedi, P. K., [TE] **etregress**, [TE] **teffects aipw**, [TE] **teffects intro advanced**, [TE] **teffects ra**

Tsiatis, A. A., [TE] **teffects aipw**, [TE] **teffects intro advanced**

U

unconfoundedness, see conditional-independence
assumption

V

van der Laan, M. J., [TE] **teffects intro advanced**

van der Vaart, A. W., [TE] **teffects aipw**

variance, Huber/White/sandwich estimator, see robust,
Huber/White/sandwich estimator of variance

Vittinghoff, E., [TE] **teffects intro advanced**

W

Walstrum, T., [TE] **etregress**

White/Huber/sandwich estimator of variance, see robust,
Huber/White/sandwich estimator of variance

Wooldridge, J. M., [TE] **etregress**, [TE] **teffects aipw**,
[TE] **teffects intro advanced**, [TE] **teffects
multivalued**, [TE] **teffects ra**

Z

Zhao, L. P., [TE] **teffects intro advanced**

