Package ‘WeightIt’

October 17, 2017

Type Package
Title Weighting for Covariate Balance in Observational Studies
Version 0.1.0
Author Noah Greifer [aut, cre]
Maintainer Noah Greifer <noah@unc.edu>
Depends R (>= 3.4.0)
Imports cobalt (>= 3.0.0)
Suggests twang (>= 1.5), CBPS (>= 0.14), ebal (>= 0.1-6), ATE (>= 0.2.0), mlogit (>= 0.2-4), MNP (>= 3.1-0)
License GPL (>= 2)
Encoding UTF-8
LazyData true
Date 2017-10-16
NeedsCompilation no
Repository CRAN
Date/Publication 2017-10-17 11:06:27 UTC

R topics documented:

summary.weightit .................................................. 2
weightit .............................................................. 3
summary.weightit

Description

`summary.weightit()` generates a summary of the `weightit` object to evaluate the properties of the estimated weights.

Usage

```r
## S3 method for class 'weightit'
summary(object, top = 5,
         ignore.s.weights = FALSE, ...)

## S3 method for class 'summary.weightit'
print(x, ...)
```

Arguments

- `object` a `weightit` object; the output of a call to `weightit()`.
- `top` how many of the largest and smallest weights to display. Default is 5.
- `ignore.s.weights` whether or not to ignore sampling weights when computing the weight summary. If `FALSE`, the default, the estimated weights will be multiplied by the sampling weights (if any) before values are computed.
- `x` a `summary.weightit` object; the output of a call to `summary.weightit()`.
- `...` arguments passed to `print`.

Value

A `summary.weightit` object with the following elements:

- `weight.range` The range (minimum and maximum) weight for each treatment group.
- `weight.top` The units with the greatest weights in each treatment group; how many are included is determined by `top`.
- `weight.ratio` The ratio of the largest weight to the smallest weight in each treatment group and overall.
- `coef.of.var` The coefficient of variation (standard deviation divided by mean) of the weights in each treatment group and overall.
- `effective.sample.size` The effective sample size for each treatment group before and after weighting.
weightit

Author(s)
Noah Greifer <noah@unc.edu>

References

See Also
weightit, summary

Examples

# See example at ?weightit.

weightit Generate Balancing Weights

Description
weightit() allows for the easy generation of balancing weights using a variety of available methods for binary, continuous, and multinomial treatments. Many of these methods exist in other packages, which weightit() calls; these packages must be installed to use the desired method. Also included are print and summary methods for examining the output.

Usage

weightit(formula, data, method, estimand = "ATE", stabilize = FALSE, focal = NULL, exact = NULL, s.weights = NULL, ps = NULL, verbose = FALSE, ...)

## S3 method for class 'weightit'
print(x, ...)

Arguments

formula a formula with a treatment variable on the left hand side and the covariates to be balanced on the right hand side. See glm for more details. Interactions and functions of covariates are allowed.
data a data set in the form of a data frame that contains the variables in formula
method a string of length 1 containing the name of the method that will be used to estimate weights. See Details below for allowable options. The default is "ps".
estimand  the desired estimand. For binary treatments, can be "ATE", "ATT", "ATC", and, for some methods, "ATO". For multinomial treatments, can be "ATE" or "ATT". The default for both is "ATE". This argument is ignored for continuous treatments.

stabilize  logical; whether or not to stabilize the weights. For the "ps" method, this involves multiplying each unit’s weight by the marginal probability of being in its treatment group. For the "ebal" method, this involves using `ebalance.trim()` to reduce the variance of the weights. Default is FALSE.

focal  when multinomial treatments are used and the "ATT" is requested, which group to consider the "treated" or focal group. This group will not be weighted, and the other groups will be weighted to be more like the focal group.

exact  a vector or the names of variables in data for which weighting is to be done within categories. For example, if exact = "gender", weights will be generated separately within each level of the variable "gender".

s.weights  A vector of sampling weights or the name of a variable in data that contains sampling weights. These are ignored for some methods. These can also be matching weights if weighting is to be used on matched data.

ps  A vector of propensity scores or the name of a variable in data containing propensity scores. If not NULL, method is ignored, and the propensity scores will be used to create weights. `formula` must include the treatment variable in data, but the listed covariates will play no role in the weight estimation.

verbose  whether to print additional information output but the fitting function.

...  other arguments for functions called by weightit that control aspects of fitting that are not covered by the above arguments. See Details.

x  a `weightit` object; the output of a call to `weightit()`.

Details

The primary purpose of `weightit()` is as a dispatcher to other functions in other packages that perform the estimation of balancing weights. These functions are identified by a name, which is used in `method` to request them. Each method has some slight distinctions in how it is called, but in general, simply entering the method will cause `weightit()` to generate the weights correctly using the function. To use each method, the package containing the function must be installed, or else an error will appear. Below are the methods allowed and some information about each.

"ps" Propensity score weighting using GLM. For binary treatments, this method estimates the propensity scores using `glm()`. An additional argument is `link`, which uses the same options as `link` in `family`. The default link is "logit", but others, including "probit", are allowed. The weights for the ATE, ATT, and ATC are computed from the estimated propensity scores using the standard formulas, and the weights for the ATO are computed as in Li, Morgan, & Zaslavsky (2016). For multinomial treatments, the propensity scores are estimated using multinomial regression from one of two functions depending on the requested link: for logit ("logit") and probit ("probit") links, `mlogit()` from the `mlogit` package is used, and for the Bayesian probit ("bayes.probit") link, `mnp()` from the `MNP` package is used. These are the only three links allowed for multinomial treatments at this time. (These methods can fail to converge, yielding errors that may seem foreign.) For continuous treatments, the generalized propensity score is estimated using linear regression with a normal density, but other families
and links are allowed, such as poisson for count data, using the family and link arguments. An additional argument, num.formula, may be specified, containing the stabilization variables on the right hand side. For all treatment types except multinomial treatments with a Bayesian probit link, sampling weights are supported, but a warning message from glm() may appear.

"gbm" Propensity score weighting using generalized boosted modeling. This method, which can also be requested as "gbr" or "twang", uses functions from the twang package to perform generalized boosted modeling to estimate propensity scores that yield balance on the requested covariates. For binary treatments, ps() is used, and the ATE, ATT, and ATC can be requested. For multinomial treatments, mmps() is used, and the ATE or ATT can be requested. For both, the weightit() argument s.weights corresponds to the ps() and mmps() argument sampw. The weightit() argument focal corresponds to the mmps() argument treatATT. For both, a single stop method must be supplied to stop.method; only one can be entered at a time. The other arguments to ps() and mmps can be specified in the call to weightit(). See ps and mmps for details.

"cbps" Covariate Balancing Propensity Score weighting. This method uses the CBPS() function from the CBPS package to estimate propensity scores and weights. It works with binary, multinomial, and continuous treatments. For binary treatments, the ATE, ATT, and ATC can be requested. For multinomial treatments, only the ATE can be requested. The weightit() argument s.weights corresponds to the CBPS() argument sampling.weights. CBPS() can fit either an over-identified model or a model that only contains covariate balancing conditions; this option is typically specified with the method argument to CBPS(), but because this argument is already used in weightit(), a new argument, over, can be specified. over = FALSE in weightit() is equivalent to method = "exact" in CBPS(). The other arguments to CBPS() can be specified in the call to weightit(). See CBPS for details.

"npcbps" Non-parametric Covariate Balancing Propensity Score weighting. This method uses the npCBPS() function from the CBPS package to estimate weights. It works with binary, multinomial, and continuous treatments. For binary and multinomial treatments, only the ATE can be requested. Sampling weights are not supported. The other arguments to npCBPS() can be specified in the call to weightit(). See npCBPS for details.

"ebal" Entropy balancing. This method uses the ebalance() function from the ebal package to estimate weights. It works with binary and multinomial treatments. For binary treatments, the ATE, ATT, and ATC can be requested. For multinomial treatments, the ATE and ATT can be requested. If the ATT is requested with a multinomial treatment, one treatment level must be entered to focal to serve as the "treated". Sampling weights are supported and are automatically entered into base.weight in ebal(). When stabilize = TRUE, ebalance.trim() is used to trim and reduce the variance of the weights. The other arguments to ebalance() can be specified in the call to weightit(). See ebalance for details.

"ebcw" Empirical balancing calibration weighting. This method uses the ATE() function from the ATE package to estimate weights. It works with binary and multinomial treatments. For binary treatments, the ATE, ATT, and ATC can be requested. For multinomial treatments, only the ATE can be requested. Sampling weights are not supported. The other arguments to ATE() can be specified in the call to weightit(). See ATE for details.

Value

A weightit object with the following elements:

weights The estimated weights, one for each unit.
treat  The values of the treatment variable.
covs   The covariates used in the fitting. Only includes the raw covariates, which may have been altered in the fitting process.
data  The data.frame originally entered to weightit().
estimand The estimand requested.
method The weight estimation method specified.
ps    The estimated or provided propensity scores.
s.weights The provided sampling weights.
discarded Unused.
treat.type The type of treatment: binary, continuous, or multinomial ("multi").
focal  The focal variable if the ATT was requested with a multinomial treatment.

Author(s)
Noah Greifer <noah@unc.edu>

References

Examples
library("cobalt")
data("lalonde", package = "cobalt")

#Balancing covariates between treatment groups
(W1 <- weightit(treat ~ age + educ + married +
                nodegree + re74, data = lalonde,
                method = "ps", estimand = "ATT"))
summary(W1)
bal.tab(W1)

#Balancing covariates among races (3-level)
(W2 <- weightit(race ~ age + educ + married +
                nodegree + re74, data = lalonde,
                method = "cbps", estimand = "ATE",
                over = FALSE))
summary(W2)
bal.tab(W2)

#Balancing covariates with respect to re78 (continuous)
(W3 <- weightit(re78 ~ age + educ + married +
                nodegree + re74, data = lalonde,
                method = "ps", num.formula = ~ age +
                educ + married))
summary(W3)
bal.tab(W3)
Index

ATE, 5
CBPS, 5
ebalance, 5
family, 4
glm, 3
mnps, 5
npCBPS, 5
print, 2
print.summary.weightit
   (summary.weightit), 2
print.weightit (weightit), 3
ps, 5
summary, 3
summary.weightit, 2
weightit, 3, 3