Package ‘twangMediation’

October 14, 2022

Title  Twang Causal Mediation Modeling via Weighting

Version  1.2

Author  Dan McCaffrey [aut, cre],
        Katherine Castellano [aut],
        Donna Coffman [aut],
        Brian Vegetabile [aut],
        Megan Schuler [aut],
        Haoyu Zhou [aut]

Maintainer  Dan McCaffrey <dmccaffrey@ets.org>


Depends  R (>= 2.10)

Imports  twang, gbm (>= 1.5-3), graphics, stats, lattice, latticeExtra, survey

Suggests  knitr, xgboost

VignetteBuilder  knitr

License  GPL-3

Encoding  UTF-8

RoxygenNote  7.1.1

LazyData  true

NeedsCompilation  no

Repository  CRAN

Date/Publication  2022-08-16 11:50:02 UTC
Description

Provides functions for estimating natural direct and indirect effects for mediation analysis. It uses weighting where the weights are functions of estimates of the probability of exposure or treatment assignment (Hong, G (2010). <https://cepa.stanford.edu/sites/default/files/workshops/GH_JSM

Author(s)

Maintainer: Dan McCaffrey <dmccaffrey@ets.org>

Authors:

- Katherine Castellano <kecastellano@ets.org>
- Donna Coffman <donna.coffman@temple.edu>
- Brian Vegetabile <bvegetab@rand.org>
- Megan Schuler <mschuler@rand.org>
- Haoyu Zhou <haoyu.zhou@temple.edu>
bal.table.mediation  

**Description**

Compute the balance table for mediation object.

**Usage**

```
bal.table.mediation(x, digits = 3, details = FALSE, plot = FALSE, ...)
```

**Arguments**

- `x`  
  A mediation object

- `digits`  
  Number of digits to round to. Default: 3

- `details`  
  logical. If `TRUE` covariate balance for the models used to create the inputs into the weights used in effect estimation is checked. If `FALSE` the additional balance is not checked.

- `plot`  
  logical. If `TRUE`, plots of the balance for all covariates are outputted for each type of effect (NIE0, NIE1, NDE0, NDE1, TE) for each stopping method. If `FALSE`, no plots are returned.

- `...`  
  Additional arguments.

**Value**

- `res`  
  tables detailing covariate balance across exposure groups both before and after weighting

**See Also**

- `print.bal.table.mediation`, `wgtmed`

**Examples**

```
data("tMdat")

## tMdat is small simulated data set included in twangMedRiation for demonstrating the functions. See ?tMdat for details

fit.es.max <- wgtmed(M ~ w1 + w2 + w3,
                     data = tMdat,
                     a_treatment = "A",
                     y_outcome = "Y",
                     total_effect_wts = tMdat$te.wgt,
                     method = "logistic"
                     )

bal.table.mediation(fit.es.max)
```
calculate_effects  
*Calculate the actual effects*

Description

Calculate the actual effects

Usage

```
calculate_effects(w_11, w_00, w_10, w_01, y_outcome, sampw = NULL)
```

Arguments

- `w_11`  The Y(1, M(1)) weights
- `w_00`  The Y(0, M(0)) weights
- `w_10`  The Y(1, M(0)) weights
- `w_01`  The Y(0, M(1)) weights
- `y_outcome`  The Y variable
- `sampw`  Sampling weights, set to NULL by default.

Value

`res`  The actual effects

See Also

`wgtmed`

check_missing  
*Check vector for NA or NAN values.*

Description

`check_missing` raises an error if the data contains NA or NAN values.

Usage

```
check_missing(x)
```

Arguments

- `x`  numeric The data set to check for NA or NAN values.

Value

Indicator of the existence of NA or NAN values
desc.effects

Describe the effects

Description

Describe the effects, and calculate standard errors and confidence intervals

Usage

desc.effects(x, ...)

Arguments

x 
An object

... 
list, optional Additional arguments.

Value

Effects, standard errors and confidence intervals of an object

See Also

desc.effects.mediation, wgtmed

Examples

data("tMdat")

## tMdat is small simulated data set included in twangMediation for
## demonstrating the functions. See ?tMdat for details

fit.es.max <- wgtmed(M ~ w1 + w2 + w3,
   data = tMdat,
   a_treatment = "A",
   y_outcome = "Y",
   total_effect_wts = tMdat$te.wgt,
   method = "ps",
   ps_n.trees=1500,
   ps_shrinkage=0.01,
   ps_stop.method=c("es.max")
)

desc.effects(fit.es.max)
desc.effects.mediation

Describe the effects from a mediation object

Description

Describe the effects, and calculate standard errors and confidence intervals from a mediation object

Usage

## S3 method for class 'mediation'
desc.effects(x, y_outcome = NULL, ...)

Arguments

x A mediation object
y_outcome The outcome; if NULL, then Y must have been provided to the original mediation function.
... Additional arguments..

Value

results effects, standard errors, and confidence intervals of a mediation object

See Also
desc.effects.wgtmed

effects.wgtmed

Examples

data("tMdat")

## tMdat is small simulated data set included in twangMediation for
## demonstrating the functions. See ?tMdat for details

fit.es.max <- wgtmed(M ~ w1 + w2 + w3,
data = tMdat,
a_treatment = "A",
y_outcome = "y",
total_effect_wts = tMdat$te.wgt,
method = "ps",
ps_n.trees=1500,
ps_shrinkage=0.01,
ps_stop.method="es.max"
)

desc.effects(fit.es.max)
dx.wts.mediation

Compute diagnostics assessing covariates balance.

Description

`dx.wts.mediation` takes a ps object or a set of propensity scores and computes diagnostics assessing covariates balance.

Usage

```r
dx.wts.mediation(
  x,  
  data,  
  estimand,  
  vars = NULL,  
  treat.var,  
  x.as.weights = TRUE,  
  sampw = NULL,  
  perm.test.iters = 0
)
```

Arguments

- `x` A data frame, matrix, or vector of propensity score weights or a ps object. `x` can also be a data frame, matrix, or vector of propensity scores if `x.as.weights=FALSE`.
- `data` A data frame.
- `estimand` The estimand of interest: either "ATT" or "ATE".
- `vars` A vector of character strings naming variables in `data` on which to assess balance.
- `treat.var` A character string indicating which variable in `data` contains the 0/1 treatment group indicator.
- `x.as.weights` TRUE or FALSE indicating whether `x` specifies propensity score weights or propensity scores. Ignored if `x` is a ps object. Default: TRUE.
- `sampw` Optional sampling weights. If `x` is a ps object, then the sampling weights should have been passed to `ps` and not specified here. `dx.wts.mediation` will issue a warning if `x` is a ps object and `sampw` is also specified.
- `perm.test.iters` A non-negative integer giving the number of iterations of the permutation test for the KS statistic. If `perm.test.iters=0`, then the function returns an analytic approximation to the p-value. This argument is ignored if `x` is a ps object. Setting `perm.test.iters=200` will yield precision to within 3% if the true p-value is 0.05. Use `perm.test.iters=500` to be within 2%.
Details

Create a balance table that compares unweighted and weighted means and standard deviations, computes effect sizes, and KS statistics to assess the ability of the propensity scores to balance the treatment and control groups.

Value

Returns a list containing

- `treat` The vector of 0/1 treatment assignment indicators.

See Also

`wgtmed`, `bal.table.mediation`, `print.mediation`, `summary.mediation`

Examples

data("tMdat")

```r
## tMdat is small simulated data set included in twangMediation for
## demonstrating the functions. See ?tMdat for details

fit.es.max <- wgtmed(M ~ w1 + w2 + w3,
                     data = tMdat,
                     a_treatment = "A",
                     y_outcome = "Y",
                     total_effect_wts = tMdat$te.wgt,
                     method = "ps",
                     ps_n.trees=1500,
                     ps_shrinkage=0.01,
                     ps_stop.method=c("es.max")
)

## dx.wts.mediation is used internally by bal.table.mediation,
## print.mediation, and summary.mediation
summary(fit.es.max)
```

---

**NSDUH_female**

A dataset containing the substance use condition and sexual orientation of 40293 women respondents to the 2017 & 2018 National Survey of Drug Use and Health.

Description

A dataset containing the substance use condition and sexual orientation of 40293 women respondents to the 2017 & 2018 National Survey of Drug Use and Health.
Usage

NSDUH_female

Format

A data frame with 40293 rows and 24 variables:

cigmon  individual smoked any cigarettes within the past month, yes or no
educ   education level, 1 = less than high school diploma, 2 = high school diploma, 3 = some college/associates degree, 4 = college degree or higher
income income level, 1 <= $20,000, 2 = $20,000 - $49,999, 3 = $50,000 - 70,000, 4 = $75,000+
NSDUHwt NSDUH sampling weight
vestr  NSDUH strata variable
verep  NSDUH replicate within stratum
employ employment status, 1 = full-time employment, 2 = part-time employment, 3 = student, 4 = unemployed, 5 = other
race   1 = non-Hispanic white, 2 = non-Hispanic Black, 3 = student, 4 = multiracial/other race
alc15 initiated alcohol use prior to 15 years old
cig15 initiated smoking prior to 15 years old, yes or no
lgb_flag 1 = lesbian, gay or sexual, 0 = heterosexual
alc_cig_depend individual meets criteria for either past-year alcohol use disorder or nicotine dependence
weight2y NSDUH sampling weights(scaled for pooling 2017 and 2018 survey years)
age age, 1 = 18-25, 2 = 26-34, 3 = 35-49, 4 = 50+

Value

NSDUH_female  A sample data for demonstration

Source

https://nsduhweb.rti.org/respweb/homepage.cfm

See Also

wgtmed

Examples

## Not run:
data(NSDUH_female)

## End(Not run)
Description

Plot the mediation object.

Usage

## S3 method for class 'mediation'
plot(x, subset = NULL, color = TRUE, ...)

Arguments

x
weighted_mediation object

subset
Used to restrict which of the stop.methods will be used in the figure. For example subset = c(1,3) would indicate that the first and third stop.methods (in alphabetical order of those specified in the original call to the mediation function) should be included in the figure. If x$method = logistic or crossval, there is no need to subset as there is only one method used.

color
If color = FALSE, figures will be gray scale. Default: TRUE.

...
Additional arguments.

Value

Distribution plots of NIE1 (distribution of mediator for treatment sample weighted to match distribution of mediator under control for the population) and NIE0 (distribution of mediator for control sample weighted to match distribution of mediator under treatment for the population) for each mediator. For continuous mediators, distributions are plotted with density curves and for categorical (factor) mediators, distributions are plotted with barplots.

See Also

wgtmed for function input

Examples

data("tMdat")

## tMdat is small simulated data set included in twangMediation for
## demonstrating the functions. See ?tMdat for details

fit.es.max <- wgtmed(M ~ w1 + w2 + w3,
                     data = tMdat,
                     a_treatment = "A",
                     y_outcome = "Y",
                     total_effect_wts = tMdat$te.wgt,
print.bal.table.mediation

method = "ps",
ps_n.trees=1500,
ps.shrinkage=0.01,
ps.stop.method=c("es.max")
)

plot(fit.es.max)

print.bal.table.mediation

Default print statement for mediation class

Description

Default print statement for mediation class

Usage

## S3 method for class 'bal.table.mediation'
print(x, ...)

Arguments

x A bal.table.mediation object.
...
Additional arguments.

Value

Default print statement.

See Also

bal.table.mediation, wgtmed

Examples

data("tMdat")

## tMdat is small simulated data set included in twangMediation for
## demonstrating the functions. See ?tMdat for details

fit.es.max <- wgtmed(M ~ w1 + w2 + w3,
data = tMdat,
a_treatment = "A",
y_outcome = "y",
total_effect_wts = tMdat$te.wgt,
method = "ps",
ps.n.trees=1500,
ps.shrinkage=0.01,
```
ps_stop.method=c("es.max")

bal.table.mediation(fit.es.max)
```

---

**Description**

Default print statement for `mediation` class

**Usage**

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

**Arguments**

- `x`: A mediation object.
- `...`: Additional arguments.

**Value**

Default print statement.

**See Also**

`wgtmed` for input.

**Examples**

```
data("tMdat")

## tMdat is small simulated data set included in twangMediation for
## demonstrating the functions. See ?tMdat for details

fit.es.max <- wgtmed(M ~ w1 + w2 + w3,
                     data = tMdat,
                     a_treatment = "A",
                     y_outcome = "y",
                     total_effect_wts = tMdat$te.wgt,
                     method = "ps",
                     ps_n.trees=1500,
                     ps_shrinkage=0.01,
                     ps_stop.method=c("es.max")
)

print(fit.es.max)
```
**summary.mediation**

Displays a useful description of a mediation object.

### Description

Displays a useful description of a mediation object.

### Usage

```r
## S3 method for class 'mediation'
summary(object, ...)
```

### Arguments

- `object` : A mediation object
- `...` : Additional arguments.

### Value

- `ps_tables` : Table of observations' propensity scores
- `mediator_distribution_check` : balance tables for NIE_1 and NIE_0

### See Also

`wgtmed`

### Examples

```r
data("tMdat")

## tMdat is small simulated data set included in twangMediation for
## demonstrating the functions. See ?tMdat for details

## The tMdat data contains the following variables
## See ?tMdat for details

fit.es.max <- wgtmed(M ~ w1 + w2 + w3,
    data = tMdat,
    a_treatment = "A",
    y_outcome = "Y",
    total_effect_wts = tMdat$te.wgt,
    method = "ps",
    ps_n.trees=1500,
    ps_shrinkage=0.01,
    ps_stop.method=c("es.max")
)

summary(fit.es.max)
```
### swapTxCtrl

*Auxiliary function to swap treatment and control in one element of a desc object from a ps object or dx.wts object*

**Description**

Call this in the wgtmed() function and the bal.table.mediation() function.

**Usage**

```r
swapTxCtrl(dd)
```

**Arguments**

- `dd` numeric An element of a desc object from a ps or dx.wts object

**Value**

A desc object with swapped treatment and control

**See Also**

- `ps`, `dx.wts.mediation`

---

### tMdat

*Simulated data for twangMediation*

**Description**

A simulate dataset for demonstrating the functions in the twangMediation package.

**Usage**

```r
tMdat
```

**Format**

A data frame with 500 rows and 7 variables:

- `w1` Simulated continuous covariate
- `w2` Simulated continuous covariate
- `w3` Simulated continuous covariate
- `A` Simulated dichotomous exposure indicator
- `Y` Simulated continuous outcome
- `M` Simulated mediator that has 11 unique values
- `te.wgt` Estimated inverse probability weight, estimated using GBM via the twang ps function
weighted_mean

Value

| tMdat | A sample of simulated data for demonstration |

See Also

wgtmed

Examples

```r
## Not run:
data(tMdat)

## End(Not run)
```

---

**weighted_mean**

*Calculate a weighted mean.*

Description

weighted_mean calculates a weighted mean, given a vector.

Usage

```r
weighted_mean(x, weights, multiplier = NULL, na.rm = TRUE)
```

Arguments

- **x**: numeric The the data set
- **weights**: numeric The weights
- **multiplier**: An additional vector to multiply Default : NULL
- **na.rm**: Whether to remove NA values. Default: TRUE

Value

numeric The weighted mean of the data.
Weighted mediation analysis.

Description

Estimate causal mediation mechanism of a treatment using propensity score weighting.

Usage

```r
wgtmed(
  formula.med,
  data,
  a_treatment,
  y_outcome = NULL,
  med_interact = NULL,
  total_effect_wts = NULL,
  total_effect_ps = NULL,
  total_effect_stop_rule = NULL,
  method = "ps",
  sampw = NULL,
  ps_n.trees = 10000,
  ps_interaction.depth = 3,
  ps_shrinkage = 0.01,
  ps_bag.fraction = 1,
  ps_n.minobsinnode = 10,
  ps_perm.test.iters = 0,
  ps_verbose = FALSE,
  ps_stop.method = c("ks.mean", "ks.max"),
  ps_version = "gbm",
  ps_ks.exact = NULL,
  ps_n.keep = 1,
  ps_n.grid = 25,
  ps_cv.folds = 10,
  ps_keep.data = FALSE
)
```

Arguments

- `formula.med`: A `formula` object relating the mediator(s) to the covariates (potential confounding variables).
- `data`: A `data.frame` dataset that includes the treatment indicator, mediator(s), and covariates.
- `a_treatment`: The (character) name of the treatment variable, which must be dichotomous (0, 1).
- `y_outcome`: The (character) name of the outcome variable, `y`. If this is not provided, then no effects will be calculated and a warning will be raised. Default: `NULL`. 
**med_interact**
The (character) vector of names of variables specified on the right-hand side of `formula.med` that consist of crossproducts or interactions between a covariate and the mediator. See the tutorial for details on these variables.

**total_effect_wts**
A vector of total effect weights, which if left NULL then `total_effect_ps` must be supplied. Default: NULL.

**total_effect_ps**
A `ps` object that contains the total effect weights.

**total_effect_stop_rule**
The stopping rule (`ks.mean`, `ks.max`, `es.mean`, `es.max`) for the total effect weights, which only needs to be specified if `total_effect_ps` is provided. Default: NULL.

**method**
The method for getting weights ("ps", "logistic", or "crossval"). Default: "ps".

**sampw**
Optional sampling weights Default: NULL.

**ps_n.trees**
Number of gbm iterations passed on to `gbm`. Default: 10000.

**ps_interaction.depth**
A positive integer denoting the tree depth used in gradient boosting. Default: 3.

**ps_shrinkage**
A numeric value between 0 and 1 denoting the learning rate. See `gbm` for more details. Default: 0.01.

**ps_bag.fraction**
A numeric value between 0 and 1 denoting the fraction of the observations randomly selected in each iteration of the gradient boosting algorithm to propose the next tree. See `gbm` for more details. Default: 1.0.

**ps_n.minobsinnode**
An integer specifying the minimum number of observations in the terminal nodes of the trees used in the gradient boosting. See `gbm` for more details. Default: 10.

**ps_perm.test.iters**
A non-negative integer giving the number of iterations of the permutation test for the KS statistic. If `perm.test.iters`=0 then the function returns an analytic approximation to the p-value. Setting `perm.test.iters`=200 will yield precision to within 3% if the true p-value is 0.05. Use `perm.test.iters`=500 to be within 2%. Default: 0.

**ps_verbose**
If TRUE, lots of information will be printed to monitor the the progress of the fitting. Default: FALSE.

**ps_stop.method**
A method or methods of measuring and summarizing balance across pretreatment variables. Current options are `ks.mean`, `ks.max`, `es.mean`, and `es.max`. `ks` refers to the Kolmogorov-Smirnov statistic and `es` refers to standardized effect size. These are summarized across the pretreatment variables by either the maximum (.max) or the mean (.mean). Default: c("ks.mean", "ks.max").

**ps_version**
"gbm", "xgboost", or "legacy", indicating which version of the twang package to use.
- "gbm" uses gradient boosting from the `gbm` package.
- "xgboost" uses gradient boosting from the `xgboost` package.
- "legacy" uses the prior implementation of the `ps` function.
ps_ks.exact  NULL or a logical indicating whether the Kolmogorov-Smirnov p-value should be based on an approximation of exact distribution from an unweighted two-sample Kolmogorov-Smirnov test. If NULL, the approximation based on the exact distribution is computed if the product of the effective sample sizes is less than 10,000. Otherwise, an approximation based on the asymptotic distribution is used. **Warning:** setting ks.exact = TRUE will add substantial computation time for larger sample sizes. Default: NULL.

ps_n.keep  A numeric variable indicating the algorithm should only consider every n.keep-th iteration of the propensity score model and optimize balance over this set instead of all iterations. Default: 1.

ps_n.grid  A numeric variable that sets the grid size for an initial search of the region most likely to minimize the stop.method. A value of n.grid=50 uses a 50 point grid from 1:n.trees. It finds the minimum, say at grid point 35. It then looks for the actual minimum between grid points 34 and 36. If specified with n.keep>1, n.grid corresponds to a grid of points on the kept iterations as defined by n.keep. Default: 25.

ps_cv.folds  A numeric variable that sets the number of cross-validation folds if using method='crossval'. Default: 10.

ps_keep.data  A logical variable that determines if the dataset should be saved in the resulting ps model objects. Default: FALSE.

Details

For users comfortable with ps, any options prefaced with ps_ are passed directly to the ps() function. Model A is used to estimate Pr(A=1 | X) where X is the vector of background covariates specified in formula.med. If method equals "ps" model A is fit using the twang ps function with estimand= "ATE". If method equals "logistic" then model A is fit using logistic regression. If method equals "crossval" then gbm using cross-validation is used to estimate model A. Because X might include variables not used to estimate the user-provided total effect weights, model A is fit rather than using the user-provided total effect weights to derive Pr(A | X). If the user uses the same set of variables to estimate their provided total effect weights as they enter in the wgtmed function to estimate the cross-world weights and the user uses the same estimation method and arguments as specified in the wgtmed function, then the estimated model A will match the model the user used to obtain the provided total effect weights.

Value

mediation object The mediation object includes the following:

- model_a The model A ps() results.
- model_m1 The model M1 ps() results.
- model_m0 The model M0 ps() results.
- data The data set used to compute models
- stopping_methods The stopping methods passed to stop.method.
- datet stamp The date when the analysis was run.
- For each stop.method, a list with the following:
- **TE** The total effect.
- **NDE_0** The natural direct effect, holding the mediator constant at 0.
- **NIE_1** The natural indirect effect, holding the exposure constant at 1.
- **NDE_1** The natural direct effect, holding the mediator constant at 1.
- **NIE_0** The natural indirect effect, holding the exposure constant at 0.
- expected_treatment0_mediator0 $E(Y(0, M(0)))$
- expected_treatment1_mediator1 $E(Y(1, M(1)))$
- expected_treatment1_mediator0 $E(Y(1, M(0)))$
- expected_treatment0_mediator1 $E(Y(0, M(1)))$ 

- **dx.wts** A list with information for checking covariate balance of for each estimated effect. Elements are TE, NIE1, NDE0, NIE0, NDE1, with results of twang dx.wts for the covariates when weighted by weights used in the estimating the effect.

### See Also

- **ps**

### Examples

```r
data("tMdat")
## tMdat is small simulated data set included in twangMediation for
## demonstrating the functions. See ?tMdat for details

head(tMdat)
## The tMdat data contains the following variables:
## w1, w2, w3 -- Simulated covariates
## A -- Simulated dichotomous exposure indicator
## M -- Simulated discrete mediator (11 values)
## Y -- Simulated continuous outcome
## te.wgt -- Estimated inverse probability weight, estimated using
## GBM via the twang ps function

fit.es.max <- wgtmed(M ~ w1 + w2 + w3,
                      data = tMdat,
                      a_treatment = "A",
                      y_outcome = "Y",
                      total_effect_wts = tMdat$te.wgt,
                      method = "ps",
                      ps_n.trees=1500,
                      ps_shrinkage=0.01,
                      ps_stop.method=c("es.max")
                     )

fit.es.max
```
Index

* **datasets**  
  NSDUH_female, 8  
  tMdat, 14  
* **models**  
  wgtmed, 16  
* **multivariate**  
  wgtmed, 16  
* **propensity**  
  twangMediation-package, 2  
* **score**  
  twangMediation-package, 2  
bal.table.mediation, 3, 8, 11  
calculate_effects, 4  
check_missing, 4  
data.frame, 16  
desc.effects, 5, 6  
desc.effects.mediation, 5, 6  
dx.wts.mediation, 7, 14  
formula, 16  
gbm, 17, 18  
NSDUH_female, 8  
plot.mediation, 10  
print.bal.table.mediation, 3, 11  
print.mediation, 8, 12  
ps, 7, 14, 18, 19  
summary.mediation, 8, 13  
swapTxCtrl, 14  
tMdat, 14  
twang, 18  
twangMediation  
  (twangMediation-package), 2  
twangMediation-package, 2  
weighted_mean, 15  
wgtmed, 3–6, 8–13, 15, 16  
xgboost, 17  
weighted_mean, 15  
wgtmed, 3–6, 8–13, 15, 16  
xgboost, 17