Package 'CIMTx'

October 12, 2022

Type Package

Title Causal Inference for Multiple Treatments with a Binary Outcome

Version 1.2.0

Description
Different methods to conduct causal inference for multiple treatments with a binary outcome, including regression adjustment, vector matching, Bayesian additive regression trees, targeted maximum likelihood and inverse probability of treatment weighting using different generalized propensity score models such as multinomial logistic regression, generalized boosted models and super learner. For more details, see the paper by Hu et al. <doi:10.1177/0962280220921909>.

License MIT + file LICENSE

Encoding UTF-8

LazyData false

RoxygenNote 7.1.2

Imports nnet (>= 7.3-16), BART (>= 2.9), twang (>= 2.5), arm (>= 1.2-12), dplyr (>= 1.0.7), Matching (>= 4.9-11), magrittr (>= 2.0.1), WeightIt (>= 0.12.0), tmle (>= 1.5.0.2), tidyr (>= 1.1.4), stats, ggplot2 (>= 3.3.5), cowplot (>= 1.1.1), mgcv (>= 1.8-38), metR (>= 0.11.0), stringr (>= 1.4.0), SuperLearner (>= 2.0-28), foreach (>= 1.5.1), doParallel (>= 1.0.16)

NeedsCompilation no

Author Liangyuan Hu [aut],
Chenyang Gu [aut],
Michael Lopez [aut],
Jiayi Ji [aut, cre]

Maintainer Jiayi Ji <jjy2876@gmail.com>

Repository CRAN

Date/Publication 2022-06-24 07:40:02 UTC

R topics documented:

ce_estimate *Causal inference with multiple treatments using observational data*

Description

The function `ce_estimate` implements the 6 different methods for causal inference with multiple treatments using observational data.

Usage

```
ce_estimate(
  y,
  x,
  w,
  method,
  formula = NULL,
  discard = FALSE,
  estimand,
  trim_perc = NULL,
  sl_library,
  reference_trt,
  boot = FALSE,
  nboots,
  verbose_boot = TRUE,
  ndpost = 1000,
  caliper = 0.25,
  n_cluster = 5,
  ...
)
```

Arguments

`y`	A numeric vector (0, 1) representing a binary outcome.
`x`	A dataframe, including all the covariates but not treatments.
`w`	A numeric vector representing the treatment groups.
`method`	A character string. Users can selected from the following methods including `"RA"`, `"VM"`, `"BART"`, `"TMLE"`, `"IPTW-Multinomial"`, `"IPTW-GBM"`, `"IPTW-SL"`, `"RAMS-Multinomial"`, `"RAMS-GBM"`, `"RAMS-SL"`.
`formula`	A `formula` object representing the variables used for the analysis. The default is to use all terms specified in `x`.
`discard`	A logical indicating whether to use the discarding rules for the BART based methods. The default is `FALSE`.
`estimand`	A character string representing the type of causal estimand. Only `"ATT"` or `"ATE"` is allowed. When the `estimand = "ATT"`, users also need to specify the reference treatment group by setting the `reference_trt` argument.

trim_perc	A 2-vector numeric value indicating the percentile at which the inverse probability of treatment weights should be trimmed. The default is NULL.
sl_library	A character vector of prediction algorithms. A list of functions included in the SuperLearner package can be found with listWrappers.
reference_trt	A numeric value indicating reference treatment group for ATT effect.
boot	A logical indicating whether or not to use nonparametric bootstrap to calculate the 95% confidence intervals of the causal effect estimates. The default is FALSE.
nboots	A numeric value representing the number of bootstrap samples.
verbose_boot	A logical value indicating whether to print the progress of nonparametric bootstrap. The default is TRUE.
ndpost	A numeric value indicating the number of posterior draws for the Bayesian methods ("BART" and "RA").
caliper	A numeric value denoting the caliper which should be used when matching (method = "VM") on the logit of GPS within each cluster formed by K-means clustering. The caliper is in standardized units. For example, caliper = 0.25 means that all matches greater than 0.25 standard deviations of the logit of GPS are dropped. The default value is 0.25.
n_cluster	A numeric value denoting the number of clusters to form using K means clustering on the logit of GPS when method = "VM". The default value is 5.
...	Other parameters that can be passed through to functions.

Value

A summary of the effect estimates can be obtained with summary function. For VM, the output contains the number of matched individuals. For BART and discard = TRUE, the output contains number of discarded individuals. For IPTW related method and boot = FALSE, the weight distributions can be visualized using plot function. For BART and RA, the output contains a list of the posterior samples of causal estimands.

References

Hu, L., Gu, C., Lopez, M., Ji, J., & Wisnivesky, J. (2020). Estimation of causal effects of multiple treatments in observational studies with a binary outcome. *Statistical Methods in Medical Research*, **29**(11), 3218–3234.

Hu, L., Gu, C. Estimation of causal effects of multiple treatments in healthcare database studies with rare outcomes. *Health Service Outcomes Research Method* **21**, 287–308 (2021).

Sparapani R, Spanbauer C, McCulloch R Nonparametric Machine Learning and Efficient Computation with Bayesian Additive Regression Trees: The BART R Package. *Journal of Statistical Software*, **97**(1), 1-66.

Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2021). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.7. URL: https://CRAN.R-project.org/package=dplyr

Venables, W. N. & Ripley, B. D. (2002) *Modern Applied Statistics with S*. Fourth Edition. Springer, New York. ISBN 0-387-95457-0

Matthew Cefalu, Greg Ridgeway, Dan McCaffrey, Andrew Morral, Beth Ann Griffin and Lane Burgette (2021). *twang: Toolkit for Weighting and Analysis of Nonequivalent Groups*. R package version 2.5. URL:https://CRAN.R-project.org/package=twang

Noah Greifer (2021). *WeightIt: Weighting for Covariate Balance in Observational Studies*. R package version 0.12.0. URL:https://CRAN.R-project.org/package=WeightIt

Hadley Wickham (2019). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.4.0. URL:https://CRAN.R-project.org/package=stringr

Andrew Gelman and Yu-Sung Su (2020). *arm: Data Analysis Using Regression and Multilevel/Hierarchical Models*. R package version 1.11-2. URL:https://CRAN.R-project.org/package=arm

Wood, S.N. (2011) Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)* **73**(1):3-36

Eric Polley, Erin LeDell, Chris Kennedy and Mark van der Laan (2021). *SuperLearner: Super Learner Prediction*. R package version 2.0-28. URL:https://CRAN.R-project.org/package=SuperLearner

Susan Gruber, Mark J. van der Laan (2012). tmle: An R Package for Targeted Maximum Likelihood Estimation. *Journal of Statistical Software*, **51**(13), 1-35.

Jasjeet S. Sekhon (2011). Multivariate and Propensity Score Matching Software with Automated Balance Optimization: The Matching Package for R. *Journal of Statistical Software*, **42**(7), 1-52

H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016.

Claus O. Wilke (2020). *cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2'*. R package version 1.1.1. URL:https://CRAN.R-project.org/package=cowplot

Elio Campitelli (2021). *metR: Tools for Easier Analysis of Meteorological Fields*. R package version 0.11.0. URL:https://github.com/eliocamp/metR

Hadley Wickham (2021). *tidyr: Tidy Messy Data*. R package version 1.1.4. *https://CRAN.R-project.org/package=tidyr*

Microsoft Corporation and Steve Weston (2020). *doParallel: Foreach Parallel Adaptor for the 'parallel' Package*. R package version 1.0.16. URL:https://CRAN.R-project.org/package=doParallel

Microsoft and Steve Weston (2020). *foreach: Provides Foreach Looping Construct. R package version 1.5.1*. URL:https://CRAN.R-project.org/package=foreach

Examples

```
lp_w_all <-
  c(
    ".4*x1 + .1*x2  - .1*x4 + .1*x5", # w = 1
    ".2 * x1 + .2 * x2  - .2 * x4 - .3 * x5"
  ) # w = 2
nlp_w_all <-
  c(
    "-.5*x1*x4  - .1*x2*x5", # w = 1
    "-.3*x1*x4 + .2*x2*x5"
  ) # w = 2
lp_y_all <- rep(".2*x1 + .3*x2 - .1*x3 - .1*x4 - .2*x5", 3)
nlp_y_all <- rep(".7*x1*x1  - .1*x2*x3", 3)
X_all <- c(
```

```
  "rnorm(0, 0.5)", # x1
  "rbeta(2, .4)", # x2
  "runif(0, 0.5)", # x3
  "rweibull(1,2)", # x4
  "rbinom(1, .4)" # x5
)

set.seed(111111)
data <- data_sim(
  sample_size = 300,
  n_trt = 3,
  x = X_all,
  lp_y = lp_y_all,
  nlp_y = nlp_y_all,
  align = FALSE,
  lp_w = lp_w_all,
  nlp_w = nlp_w_all,
  tau = c(-1.5, 0, 1.5),
  delta = c(0.5, 0.5),
  psi = 1
)
ce_estimate(
  y = data$y, x = data$covariates, w = data$w,
  ndpost = 100, method = "RA", estimand = "ATE"
)
```

`ce_estimate_bart_ate` *Causal inference with multiple treatments using BART for ATE effects*

Description

The function `ce_estimate_bart_ate` implements BART to estimate ATE effect with multiple treatments using observational data.

Usage

```
ce_estimate_bart_ate(y, x, w, discard = FALSE, ndpost = 1000, ...)
```

Arguments

`y`	A numeric vector (0, 1) representing a binary outcome.
`x`	A dataframe, including all the covariates but not treatments.
`w`	A numeric vector representing the treatment groups.
`discard`	A logical indicating whether to use the discarding rules. The default is `FALSE`.
`ndpost`	A numeric value indicating the number of posterior draws.
`...`	Other parameters that can be passed through to functions.

Value

A summary of the effect estimates can be obtained with `summary` function. The output also contains a list of the posterior samples of causal estimands. When `discard = TRUE`, the output contains number of discarded individuals.

References

Sparapani R, Spanbauer C, McCulloch R Nonparametric Machine Learning and Efficient Computation with Bayesian Additive Regression Trees: The BART R Package. *Journal of Statistical Software*, **97**(1), 1-66.

Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2021). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.7. URL: https://CRAN.R-project.org/package=dplyr

---

`ce_estimate_bart_att` *Causal inference with multiple treatments using BART for ATT effects*

Description

The function `ce_estimate_bart_att` implements BART to estimate ATT effect with multiple treatments using observational data.

Usage

```
ce_estimate_bart_att(
  y,
  x,
  w,
  discard = FALSE,
  ndpost = 1000,
  reference_trt,
  ...
)
```

Arguments

`y`	A numeric vector (0, 1) representing a binary outcome.
`x`	A dataframe, including all the covariates but not treatments.
`w`	A numeric vector representing the treatment groups.
`discard`	A logical indicating whether to use the discarding rules. The default is `FALSE`.
`ndpost`	A numeric value indicating the number of posterior draws.
`reference_trt`	A numeric value indicating reference treatment group for ATT effect.
`...`	Other parameters that can be passed through to functions.

Value

A summary of the effect estimates can be obtained with `summary` function. The output also contains a list of the posterior samples of causal estimands. When `discard = TRUE`, the output contains number of discarded individuals.

References

Sparapani R, Spanbauer C, McCulloch R Nonparametric Machine Learning and Efficient Computation with Bayesian Additive Regression Trees: The BART R Package. *Journal of Statistical Software*, **97**(1), 1-66.

Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2021). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.7. URL: https://CRAN.R-project.org/package=dplyr

---

`ce_estimate_iptw_ate` *Causal inference with multiple treatments using IPTW for ATE effects*

Description

The function `ce_estimate_iptw_ate` implements IPTW to estimate ATE effect with multiple treatments using observational data.

Usage

```
ce_estimate_iptw_ate(y, w, x, method, ...)
```

Arguments

`y`	A numeric vector (0, 1) representing a binary outcome.
`w`	A numeric vector representing the treatment groups.
`x`	A dataframe, including all the covariates but not treatments.
`method`	A character string. Users can selected from the following methods including `"IPTW-Multinomial"`, `"IPTW-GBM"`, `"IPTW-SL"`.
`...`	Other parameters that can be passed through to functions.

Value

A summary of the effect estimates can be obtained with `summary` function. The weight distributions can be visualized using `plot` function.

References

Venables, W. N. & Ripley, B. D. (2002) *Modern Applied Statistics with S*. Fourth Edition. Springer, New York. ISBN 0-387-95457-0

Matthew Cefalu, Greg Ridgeway, Dan McCaffrey, Andrew Morral, Beth Ann Griffin and Lane Burgette (2021). *twang: Toolkit for Weighting and Analysis of Nonequivalent Groups*. R package version 2.5. URL:https://CRAN.R-project.org/package=twang

Noah Greifer (2021). *WeightIt: Weighting for Covariate Balance in Observational Studies*. R package version 0.12.0. URL:https://CRAN.R-project.org/package=WeightIt

---

`ce_estimate_iptw_ate_boot`

Causal inference with multiple treatments using IPTW for ATE effects (bootstrapping for CI)

Description

The function `ce_estimate_iptw_ate_boot` implements IPTW with bootstrapping to estimate ATE effect with multiple treatments using observational data.

Usage

```
ce_estimate_iptw_ate_boot(y, x, w, method, nboots, verbose_boot, ...)
```

Arguments

`y`	A numeric vector (0, 1) representing a binary outcome.
`x`	A dataframe, including all the covariates but not treatments.
`w`	A numeric vector representing the treatment groups.
`method`	A character string. Users can selected from the following methods including `"IPTW-Multinomial"`, `"IPTW-GBM"`, `"IPTW-SL"`.
`nboots`	A numeric value representing the number of bootstrap samples.
`verbose_boot`	A logical value indicating whether to print the progress of nonparametric bootstrap.
`...`	Other parameters that can be passed through to functions.

Value

A summary of the effect estimates can be obtained with `summary` function.

References

Hadley Wickham (2019). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.4.0. URL:https://CRAN.R-project.org/package=stringr

`ce_estimate_iptw_att` *Causal inference with multiple treatments using IPTW for ATT effects*

Description

The function `ce_estimate_iptw_att` implements IPTW to estimate ATT effect with multiple treatments using observational data.

Usage

```
ce_estimate_iptw_att(y, x, w, method, reference_trt, ...)
```

Arguments

`y`	A numeric vector (0, 1) representing a binary outcome.
`x`	A dataframe, including all the covariates but not treatments.
`w`	A numeric vector representing the treatment groups.
`method`	A character string. Users can selected from the following methods including `"IPTW-Multinomial"`, `"IPTW-GBM"`, `"IPTW-SL"`.
`reference_trt`	A numeric value indicating reference treatment group for ATT effect.
`...`	Other parameters that can be passed through to functions.

Value

A summary of the effect estimates can be obtained with `summary` function. The weight distributions can be visualized using `plot` function.

References

Venables, W. N. & Ripley, B. D. (2002) *Modern Applied Statistics with S*. Fourth Edition. Springer, New York. ISBN 0-387-95457-0

Matthew Cefalu, Greg Ridgeway, Dan McCaffrey, Andrew Morral, Beth Ann Griffin and Lane Burgette (2021). *twang: Toolkit for Weighting and Analysis of Nonequivalent Groups*. R package version 2.5. URL:https://CRAN.R-project.org/package=twang

Noah Greifer (2021). *WeightIt: Weighting for Covariate Balance in Observational Studies*. R package version 0.12.0. URL:https://CRAN.R-project.org/package=WeightIt

`ce_estimate_iptw_att_boot` *Causal inference with multiple treatments using IPTW for ATT effects (bootstrapping for CI)*

Description

The function `ce_estimate_iptw_att_boot` implements IPTW with bootstrapping to estimate ATT effect with multiple treatments using observational data.

Usage

```
ce_estimate_iptw_att_boot(
  y,
  x,
  w,
  reference_trt,
  method,
  nboots,
  verbose_boot,
  ...
)
```

Arguments

`y`	A numeric vector (0, 1) representing a binary outcome.
`x`	A dataframe, including all the covariates but not treatments.
`w`	A numeric vector representing the treatment groups.
`reference_trt`	A numeric value indicating reference treatment group for ATT effect.
`method`	A character string. Users can selected from the following methods including `"IPTW-Multinomial"`, `"IPTW-GBM"`, `"IPTW-SL"`.
`nboots`	A numeric value representing the number of bootstrap samples.
`verbose_boot`	A logical value indicating whether to print the progress of nonparametric bootstrap. The default is `TRUE`.
`...`	Other parameters that can be passed through to functions.

Value

A summary of the effect estimates can be obtained with `summary` function.

References

Hadley Wickham (2019). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.4.0. URL:https://CRAN.R-project.org/package=stringr

`ce_estimate_rams_ate` *Causal inference with multiple treatments using RAMS for ATE effects*

Description

The function `ce_estimate_rams_ate` implements RAMS to estimate ATE effect with multiple treatments using observational data.

Usage

```
ce_estimate_rams_ate(y, w, x, method, ...)
```

Arguments

`y`	A numeric vector (0, 1) representing a binary outcome.
`w`	A numeric vector representing the treatment groups.
`x`	A dataframe, including all the covariates but not treatments.
`method`	A character string. Users can selected from the following methods including `"RAMS-Multinomial"`, `"RAMS-GBM"`, `"RAMS-SL"`.
`...`	Other parameters that can be passed through to functions.

Value

A summary of the effect estimates can be obtained with `summary` function.

References

Matthew Cefalu, Greg Ridgeway, Dan McCaffrey, Andrew Morral, Beth Ann Griffin and Lane Burgette (2021). *twang: Toolkit for Weighting and Analysis of Nonequivalent Groups*. R package version 2.5. URL:https://CRAN.R-project.org/package=twang

Venables, W. N. & Ripley, B. D. (2002) *Modern Applied Statistics with S*. Fourth Edition. Springer, New York. ISBN 0-387-95457-0

Noah Greifer (2021). *WeightIt: Weighting for Covariate Balance in Observational Studies*. R package version 0.12.0. URL:https://CRAN.R-project.org/package=WeightIt

Wood, S.N. (2011) Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)* **73**(1):3-36

`ce_estimate_rams_ate_boot` *Causal inference with multiple treatments using RAMS for ATE effects (bootstrapping for CI)*

Description

The function `ce_estimate_rams_ate_boot` implements RAMS with bootstrapping to estimate ATE effect with multiple treatments using observational data.

Usage

```
ce_estimate_rams_ate_boot(y, w, x, method, nboots, verbose_boot, ...)
```

Arguments

`y`	A numeric vector (0, 1) representing a binary outcome.
`w`	A numeric vector representing the treatment groups.
`x`	A dataframe, including all the covariates but not treatments.
`method`	A character string. Users can selected from the following methods including `"RAMS-Multinomial"`, `"RAMS-GBM"`, `"RAMS-SL"`.
`nboots`	A numeric value representing the number of bootstrap samples.
`verbose_boot`	A logical value indicating whether to print the progress of nonparametric bootstrap.
`...`	Other parameters that can be passed through to functions.

Value

A summary of the effect estimates can be obtained with `summary` function.

References

Hadley Wickham (2019). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.4.0. URL:https://CRAN.R-project.org/package=stringr

`ce_estimate_rams_att` *Causal inference with multiple treatments using RAMS for ATT effects*

Description

The function `ce_estimate_rams_att` implements RAMS to estimate ATT effect with multiple treatments using observational data.

Usage

```
ce_estimate_rams_att(y, w, x, method, reference_trt, ...)
```

Arguments

`y`	A numeric vector (0, 1) representing a binary outcome.
`w`	A numeric vector representing the treatment groups.
`x`	A dataframe, including all the covariates but not treatments.
`method`	A character string. Users can selected from the following methods including `"RAMS-Multinomial"`, `"RAMS-GBM"`, `"RAMS-SL"`.
`reference_trt`	A numeric value indicating reference treatment group for ATT effect.
`...`	Other parameters that can be passed through to functions.

Value

A summary of the effect estimates can be obtained with `summary` function.

References

Matthew Cefalu, Greg Ridgeway, Dan McCaffrey, Andrew Morral, Beth Ann Griffin and Lane Burgette (2021). *twang: Toolkit for Weighting and Analysis of Nonequivalent Groups*. R package version 2.5. URL:https://CRAN.R-project.org/package=twang

Venables, W. N. & Ripley, B. D. (2002) *Modern Applied Statistics with S*. Fourth Edition. Springer, New York. ISBN 0-387-95457-0

Noah Greifer (2021). *WeightIt: Weighting for Covariate Balance in Observational Studies*. R package version 0.12.0. URL:https://CRAN.R-project.org/package=WeightIt

Wood, S.N. (2011) Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)* **73**(1):3-36

`ce_estimate_rams_att_boot` *Causal inference with multiple treatments using RAMS for ATT effects (bootstrapping for CI)*

Description

The function `ce_estimate_rams_att_boot` implements RAMS with bootstrapping to estimate ATT effect with multiple treatments using observational data.

Usage

```
ce_estimate_rams_att_boot(
  y,
  w,
  x,
  reference_trt,
  method,
  nboots,
  verbose_boot,
  ...
)
```

Arguments

`y`	A numeric vector (0, 1) representing a binary outcome.
`w`	A numeric vector representing the treatment groups.
`x`	A dataframe, including all the covariates but not treatments.
`reference_trt`	A numeric value indicating reference treatment group for ATT effect.
`method`	A character string. Users can selected from the following methods including `"RAMS-Multinomial"`, `"RAMS-GBM"`, `"RAMS-SL"`.
`nboots`	A numeric value representing the number of bootstrap samples.
`verbose_boot`	A logical value indicating whether to print the progress of nonparametric bootstrap.
`...`	Other parameters that can be passed through to functions.

Value

A summary of the effect estimates can be obtained with `summary` function.

References

Hadley Wickham (2019). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.4.0. URL:https://CRAN.R-project.org/package=stringr

ce_estimate_ra_ate *Causal inference with multiple treatments using RA for ATE effects*

Description

The function `ce_estimate_ra_ate` implements RA to estimate ATE effect with multiple treatments using observational data.

Usage

```
ce_estimate_ra_ate(y, x, w, ndpost)
```

Arguments

`y`	A numeric vector (0, 1) representing a binary outcome.
`x`	A dataframe, including all the covariates but not treatments.
`w`	A numeric vector representing the treatment groups.
`ndpost`	A numeric value indicating the number of posterior draws.

Value

A summary of the effect estimates can be obtained with `summary` function. The output also contains a list of the posterior samples of causal estimands.

References

Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2021). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.7. URL: https://CRAN.R-project.org/package=dplyr

Andrew Gelman and Yu-Sung Su (2020). *arm: Data Analysis Using Regression and Multilevel/Hierarchical Models*. R package version 1.11-2. *https://CRAN.R-project.org/package=arm*

ce_estimate_ra_att *Causal inference with multiple treatments using RA for ATT effects*

Description

The function `ce_estimate_ra_att` implements RA to estimate ATT effect with multiple treatments using observational data.

Usage

```
ce_estimate_ra_att(y, x, w, ndpost, reference_trt)
```

Arguments

y	A numeric vector (0, 1) representing a binary outcome.
x	A dataframe, including all the covariates but not treatments.
w	A numeric vector representing the treatment groups.
ndpost	A numeric value indicating the number of posterior draws.
reference_trt	A numeric value indicating reference treatment group for ATT effect.

Value

A summary of the effect estimates can be obtained with summary function. The output also contains a list of the posterior samples of causal estimands.

References

Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2021). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.7. URL: https://CRAN.R-project.org/package=dplyr

Andrew Gelman and Yu-Sung Su (2020). *arm: Data Analysis Using Regression and Multilevel/Hierarchical Models*. R package version 1.11-2. *https://CRAN.R-project.org/package=arm*

---

ce_estimate_tmle_ate *Causal inference with multiple treatments using TMLE for ATE effects*

Description

The function ce_estimate_tmle_ate implements TMLE to estimate ATE effect with multiple treatments using observational data.

Usage

```
ce_estimate_tmle_ate(y, w, x, sl_library, ...)
```

Arguments

y	A numeric vector (0, 1) representing a binary outcome.
w	A numeric vector representing the treatment groups.
x	A dataframe, including all the covariates but not treatments.
sl_library	A character vector of prediction algorithms. A list of functions included in the SuperLearner package can be found with listWrappers.
...	Other parameters that can be passed through to functions.

Value

A summary of the effect estimates can be obtained with summary function.

References

Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2021). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.7. URL: https://CRAN.R-project.org/package=dplyr

Eric Polley, Erin LeDell, Chris Kennedy and Mark van der Laan (2021). *SuperLearner: Super Learner Prediction*. R package version 2.0-28. URL:https://CRAN.R-project.org/package=SuperLearner

Susan Gruber, Mark J. van der Laan (2012). tmle: An R Package for Targeted Maximum Likelihood Estimation. *Journal of Statistical Software*, **51**(13), 1-35.

`ce_estimate_tmle_ate_boot`

Causal inference with multiple treatments using TMLE for ATE effects (bootstrapping for CI)

Description

Causal inference with multiple treatments using TMLE for ATE effects (bootstrapping for CI)

Usage

```
ce_estimate_tmle_ate_boot(y, x, w, nboots, verbose_boot, ...)
```

Arguments

`y`	A numeric vector (0, 1) representing a binary outcome.
`x`	A dataframe, including all the covariates but not treatments.
`w`	A numeric vector representing the treatment groups.
`nboots`	A numeric value representing the number of bootstrap samples.
`verbose_boot`	A logical value indicating whether to print the progress of nonparametric bootstrap.
`...`	Other parameters that can be passed through to functions.

Value

A summary of the effect estimates can be obtained with `summary` function.

References

Hadley Wickham (2019). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.4.0. URL:https://CRAN.R-project.org/package=stringr

`ce_estimate_vm_att` *Causal inference with multiple treatments using VM for ATT effects*

Description

The function `ce_estimate_vm_att` implements VM to estimate ATT effect with multiple treatments using observational data.

Usage

```
ce_estimate_vm_att(y, x, w, reference_trt, caliper, n_cluster)
```

Arguments

`y`	A numeric vector (0, 1) representing a binary outcome.
`x`	A dataframe, including all the covariates but not treatments.
`w`	A numeric vector representing the treatment groups.
`reference_trt`	A numeric value indicating reference treatment group for ATT effect.
`caliper`	A numeric value denoting the caliper on the logit of GPS within each cluster formed by K-means clustering. The caliper is in standardized units. For example, `caliper = 0.25` means that all matches greater than 0.25 standard deviations of the logit of GPS are dropped. The default value is 0.25.
`n_cluster`	A numeric value denoting the number of clusters to form using K means clustering on the logit of GPS.

Value

A summary of the effect estimates can be obtained with `summary` function. The output also contains the number of matched individuals.

References

Venables, W. N. & Ripley, B. D. (2002) *Modern Applied Statistics with S*. Fourth Edition. Springer, New York. ISBN 0-387-95457-0

Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2021). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.7. URL: https://CRAN.R-project.org/package=dplyr

Jasjeet S. Sekhon (2011). Multivariate and Propensity Score Matching Software with A utomated Balance Optimization: The Matching Package for R. *Journal of Statistical Software*, **42**(7), 1-52

covariate_overlap *Covariate overlap figure*

Description

Covariate overlap figure

Usage

```
covariate_overlap(treatment, prob)
```

Arguments

treatment	A numeric vector representing the treatment groups.
prob	Probability of being assigned to the treatment group.

Value

A ggplot figure.

References

H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016.

Claus O. Wilke (2020). *cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2'*. R package version 1.1.1. URL:https://CRAN.R-project.org/package=cowplot

data_sim *Simulate data for binary outcome with multiple treatments*

Description

The function data_sim simulate data for binary outcome with multiple treatments. Users can adjust the following 7 design factors: (1) sample size, (2) ratio of units across treatment groups, (3) whether the treatment assignment model and the outcome generating model are linear or nonlinear, (4) whether the covariates that best predict the treatment also predict the outcome well, (5) whether the response surfaces are parallel across treatment groups, (6) outcome prevalence, and (7) degree of covariate overlap.

Usage

```
data_sim(
  sample_size,
  n_trt = 3,
  x = "rnorm(0, 1)",
  lp_y = rep("x1", 3),
  nlp_y = NULL,
  align = TRUE,
  tau = c(0, 0, 0),
  delta = c(0, 0),
  psi = 1,
  lp_w,
  nlp_w
)
```

Arguments

sample_size	A numeric value indicating the total number of units.
n_trt	A numeric value indicating the number of treatments. The default is set to 3.
x	A vector of characters representing covariates, with each covariate being generated from the standard probability. The default is set to "rnorm(0, 1)". distributions in the stats package.
lp_y	A vector of characters of length n_trt, representing the linear effects in the outcome generating model. The default is set to rep("x1", 3).
nlp_y	A vector of characters of length n_trt, representing the nonlinear effects in the outcome generating model. The default is set to NULL.
align	A logical indicating whether the predictors in the treatment assignment model are the same as the predictors for the outcome generating model. The default is TRUE. If the argument is set to FALSE, users need to specify additional two arguments lp_w and nlp_w.
tau	A numeric vector of length n_trt inducing different outcome event probabilities across treatment groups. Higher values mean higher outcome event probability for the treatment group; lower values mean lower outcome event probability for the treatment group. The default is set to c(0, 0, 0), which corresponds to an approximately equal outcome event probability across three treatment groups.
delta	A numeric vector of length n_trt-1 inducing different ratio of units across treatment groups. Higher values mean higher proportion for the treatment group; lower values mean lower proportion for the treatment group. The default is set to c(0,0), which corresponds to an approximately equal sample sizes across three treatment groups.
psi	A numeric value for the parameter governing the sparsity of covariate overlap. Higher values mean weaker covariate overlap; lower values mean stronger covariate overlap. The default is set to 1, which corresponds to a moderate covariate overlap.
lp_w	is a vector of characters of length n_trt - 1, representing in the treatment assignment model

`nlp_w` is a vector of characters of length `n_trt - 1`, representing in the treatment assignment model

Value

A list with 7 elements for simulated data. It contains

`covariates:` x matrix

`w:` treatment indicators

`y:` observed binary outcomes

`y_prev:` outcome prevalence rates

`ratio_of_units:` the proportions of units in each treatment group

`overlap_fig:` the visualization of covariate overlap via boxplots of the distributions of true GPS

`y_true:` simulated true outcome in each treatment group

References

Hadley Wickham (2019). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.4.0. URL:https://CRAN.R-project.org/package=stringr

Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2021). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.7. URL: https://CRAN.R-project.org/package=dplyr

Examples

```
library(CIMTx)
lp_w_all <-
  c(
    ".4*x1 + .1*x2  - .1*x4 + .1*x5", # w = 1
    ".2 * x1 + .2 * x2  - .2 * x4 - .3 * x5"
  ) # w = 2
nlp_w_all <-
  c(
    "-.5*x1*x4  - .1*x2*x5", # w = 1
    "-.3*x1*x4 + .2*x2*x5"
  ) # w = 2
lp_y_all <- rep(".2*x1 + .3*x2 - .1*x3 - .1*x4 - .2*x5", 3)
nlp_y_all <- rep(".7*x1*x1  - .1*x2*x3", 3)
X_all <- c(
  "rnorm(0, 0.5)", # x1
  "rbeta(2,0.4)", # x2
  "runif(0, 0.5)", # x3
  "rweibull(1,2)", # x4
  "rbinom(1,0.4)" # x5
)

set.seed(111111)
```

```
data <- data_sim(
  sample_size = 300,
  n_trt = 3,
  x = X_all,
  lp_y = lp_y_all,
  nlp_y = nlp_y_all,
  align = FALSE,
  lp_w = lp_w_all,
  nlp_w = nlp_w_all,
  tau = c(-1.5, 0, 1.5),
  delta = c(0.5, 0.5),
  psi = 1
)
```

`plot.CIMTx_ATE_posterior` *Plot for non-IPTW estimation methods with bootstrapping for ATE effect*

Description

Plot for non-IPTW estimation methods with bootstrapping for ATE effect

Usage

```
## S3 method for class 'CIMTx_ATE_posterior'
plot(x, ...)
```

Arguments

`x`	a `CIMTx_ATE_posterior` object obtained
`...`	further arguments passed to or from other methods.

Value

an error message

Examples

```
## Not run:
result <- list(method = "RA")
class(result) <- "CIMTx_ATE_posterior"
plot(result)

## End(Not run)
```

plot.CIMTx_ATT_posterior *Plot for non-IPTW estimation methods for ATT effect*

Description

Plot for non-IPTW estimation methods for ATT effect

Usage

```
## S3 method for class 'CIMTx_ATT_posterior'
plot(x, ...)
```

Arguments

x	a CIMTx_ATT_posterior object obtained
...	further arguments passed to or from other methods.

Value

an error message

Examples

```
## Not run:
result <- list(method = "RA")
class(result) <- "CIMTx_ATT_posterior"
plot(result)

## End(Not run)
```

plot.CIMTx_IPTW *Boxplot for weight distribution*

Description

This function make the boxplot plot for the weights estimated by different IPTW methods. The inputs of the function are from the output of ce_estimate.R function when the methods are "IPTW-Multinomial", "IPTW-GBM", "IPTW-SL".

Usage

```
## S3 method for class 'CIMTx_IPTW'
plot(...)
```

Arguments

...	Objects from IPTW related methods

Value

A ggplot figure

References

Hadley Wickham (2019). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.4.0. URL:https://CRAN.R-project.org/package=stringr

Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2021). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.7. URL: https://CRAN.R-project.org/package=dplyr

H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016.

Claus O. Wilke (2020). *cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2'*. R package version 1.1.1. URL:https://CRAN.R-project.org/package=cowplot

Examples

```
iptw_object_example <- list(weight = rnorm(1000, 1, 1), method = "IPTW-SL")
class(iptw_object_example) <- "CIMTx_IPTW"
plot(iptw_object_example)
```

---

plot.CIMTx_nonIPTW_once

Plot for non-IPTW estimation methods for ATE effect

Description

Plot for non-IPTW estimation methods for ATE effect

Usage

```
## S3 method for class 'CIMTx_nonIPTW_once'
plot(x, ...)
```

Arguments

x	a CIMTx_nonIPTW_once object obtained
...	further arguments passed to or from other methods.

Value

an error message

Examples

```
## Not run:
result <- list(method = "TMLE")
class(result) <- "CIMTx_nonIPTW_once"
plot(result)

## End(Not run)
```

`plot.CIMTx_sa_grid` *Contour plot for the grid specification of sensitivity analysis*

Description

This function make the countor plot after the grid specification of sensitivity analysis. The input of the function is from the output of the sa.R function.

Usage

```
## S3 method for class 'CIMTx_sa_grid'
plot(x, ate = NULL, att = NULL, ...)
```

Arguments

`x`	Object from sa function
`ate`	a character indicating the ATE effect to plot, eg, "1,3" or "2,3"
`att`	a character indicating the ATT effect to plot, eg, "1,3" or "1,2"
`...`	further arguments passed to or from other methods.

Value

A ggplot figure

References

Hadley Wickham (2019). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.4.0. URL:https://CRAN.R-project.org/package=stringr

Elio Campitelli (2021). *metR: Tools for Easier Analysis of Meteorological Fields*. R package version 0.11.0. URL:https://github.com/eliocamp/metR

Examples

```
sa_object_example <- list(
  ATE13 = seq(0, 1, length.out = 25), grid_index = c(4, 5),
  c_functions = data.frame(
    c4 = rep(seq(-0.6, 0, 0.15), each = 5),
    c5 = rep(seq(0, 0.6, 0.15), 5)
  )
```

```
)
class(sa_object_example) <- "CIMTx_sa_grid"
plot(sa_object_example, ate = "1,3")
```

`posterior_summary` *Posterior distribution summary*

Description

Posterior distribution summary

Usage

```
posterior_summary(rd_est, rr_est, or_est)
```

Arguments

`rd_est`	Posterior distribution for rd estimates.
`rr_est`	Posterior distribution for rr estimates.
`or_est`	Posterior distribution for rr estimates.

Value

A list of causal estimands including risk difference (rd), odds ratios (or) and relative risk (rr) between different treatment groups.

`print.CIMTx_ATE_posterior`
Print the ATE results for non-IPTW results

Description

The `print` method for class "CIMTx_ATE_posterior"

Usage

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

Arguments

`x`	a `CIMTx_ATE_posterior` object obtained from `ce_estimate` function
`...`	further arguments passed to or from other methods.

Value

The output from print

Examples

```
result <- list(method = "RA")
class(result) <- "CIMTx_ATE_posterior"
print(result)
```

`print.CIMTx_ATE_sa` *Print the ATE results for from sensitivity analysis*

Description

The print method for class "CIMTx_ATE_sa"

Usage

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

Arguments

x	a CIMTx_ATE_sa object obtained from sa function
...	further arguments passed to or from other methods.

Value

The output from print

Examples

```
result <- list(estimand = "ATE")
class(result) <- "CIMTx_ATE_sa"
print(result)
```

print.CIMTx_ATT_posterior *Print the ATT results*

Description

The print method for class "CIMTx_ATT_posterior"

Usage

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

Arguments

x	a CIMTx_ATT_posterior x obtained from ce_estimate function
...	further arguments passed to or from other methods.

Value

The output from print

Examples

```
result <- list(method = "RA")
class(result) <- "CIMTx_ATT_posterior"
print(result)
```

print.CIMTx_ATT_sa *Print the ATT results for from sensitivity analysis*

Description

The print method for class "CIMTx_ATT_sa"

Usage

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

Arguments

x	a CIMTx_ATT_sa object obtained from sa function
...	further arguments passed to or from other methods.

Value

The output from print

Examples

```
result <- list(estimand = "ATT")
class(result) <- "CIMTx_ATT_sa"
print(result)
```

`print.CIMTx_IPTW` *Print the ATE/ATT results for IPTW results*

Description

The print method for class "CIMTx_IPTW"

Usage

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

Arguments

x	a CIMTx_ATE_posterior x obtained
...	further arguments passed to or from other methods.

Value

The output from print

Examples

```
result <- list(method = "IPTW-Multinomial", estimand = "ATE")
class(result) <- "CIMTx_IPTW"
print(result)
```

print.CIMTx_nonIPTW_once *Print the ATE/ATT results for non-IPTW results*

Description

The print method for class "CIMTx_nonIPTW_once"

Usage

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

Arguments

x	a CIMTx_ATE_posterior x obtained from ce_estimate function
...	further arguments passed to or from other methods.

Value

The output from print

Examples

```
result <- list(method = "TMLE", estimand = "ATE")
class(result) <- "CIMTx_nonIPTW_once"
print(result)
```

print.CIMTx_sa_grid *Print the ATT results for from sensitivity analysis*

Description

The print method for class "CIMTx_sa_grid"

Usage

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

Arguments

x	a CIMTx_sa_grid object obtained from sa function
...	further arguments passed to or from other methods.

Value

The output from print

Examples

```
result <- list(estimand = "ATE")
class(result) <- "CIMTx_sa_grid"
print(result)
```

---

sa *Flexible Monte Carlo sensitivity analysis for unmeasured confounding*

Description

The function sa implements the flexible sensitivity analysis approach for unmeasured confounding with multiple treatments and a binary outcome.

Usage

```
sa(
  x,
  y,
  w,
  formula = NULL,
  prior_c_function,
  m1,
  m2 = NULL,
  n_cores = 1,
  estimand,
  reference_trt,
  ...
)
```

Arguments

x	A dataframe, including all the covariates but not treatments.
y	A numeric vector (0, 1) representing a binary outcome.
w	A numeric vector representing the treatment groups.
formula	A formula object for the analysis. The default is to use all terms specified in x.
prior_c_function	1) A vector of characters indicating the prior distributions for the confounding functions. Each character contains the random number generation code from the standard probability distributions in the stats package. 2) A vector of characters including the grid specifications for the confounding functions. It should be used when users want to formulate the confounding functions as scalar values. 3) A matrix indicating the point mass prior for the confounding functions

m1	A numeric value indicating the number of draws of the GPS from the posterior predictive distribution
m2	A numeric value indicating the number of draws from the prior distributions of the confounding functions
n_cores	A numeric value indicating number of cores to use for parallel computing.
estimand	A character string representing the type of causal estimand. Only "ATT" or "ATE" is allowed. When the estimand = "ATT", users also need to specify the reference treatment group by setting the reference_trt argument.
reference_trt	A numeric value indicating reference treatment group for ATT effect.
...	Other parameters that can be passed to BART functions

Value

A list of causal estimands including risk difference (RD) between different treatment groups.

References

Hadley Wickham (2019). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.4.0. URL:https://CRAN.R-project.org/package=stringr

Hadley Wickham (2021). *tidyr: Tidy Messy Data*. R package version 1.1.4. URL:https://CRAN.R-project.org/package=tidyr

Sparapani R, Spanbauer C, McCulloch R Nonparametric Machine Learning and Efficient Computation with Bayesian Additive Regression Trees: The BART R Package. *Journal of Statistical Software*, **97**(1), 1-66.

Microsoft Corporation and Steve Weston (2020). *doParallel: Foreach Parallel Adaptor for the 'parallel' Package*. R package version 1.0.16. URL:https://CRAN.R-project.org/package=doParallel

Microsoft and Steve Weston (2020). *foreach: Provides Foreach Looping Construct.*. R package version 1.5.1 URL:https://CRAN.R-project.org/package=foreach

Examples

```
lp_w_all <-
  c(
    ".4*x1 + .1*x2  - 1.1*x4 + 1.1*x5", # w = 1
    ".2 * x1 + .2 * x2  - 1.2 * x4 - 1.3 * x5"
  ) # w = 2
nlp_w_all <-
  c(
    "-.5*x1*x4  - .1*x2*x5", # w = 1
    "-.3*x1*x4 + .2*x2*x5"
  ) # w = 2
lp_y_all <- rep(".2*x1 + .3*x2 - .1*x3 - 1.1*x4 - 1.2*x5", 3)
nlp_y_all <- rep(".7*x1*x1  - .1*x2*x3", 3)
X_all <- c(
  "rnorm(0, 0.5)", # x1
  "rbeta(2, .4)", # x2
```

```
  "runif(0, 0.5)", # x3
  "rweibull(1,2)", # x4
  "rbinom(1, .4)" # x5
)
set.seed(1111)
data <- data_sim(
  sample_size = 100,
  n_trt = 3,
  x = X_all,
  lp_y = lp_y_all,
  nlp_y = nlp_y_all,
  align = FALSE,
  lp_w = lp_w_all,
  nlp_w = nlp_w_all,
  tau = c(0.5, -0.5, 0.5),
  delta = c(0.5, 0.5),
  psi = 2
)
c_grid <- c(
  "runif(-0.6, 0)", # c(1,2)
  "runif(0, 0.6)", # c(2,1)
  "runif(-0.6, 0)", # c(2,3)
  "seq(-0.6, 0, by = 0.3)", # c(1,3)
  "seq(0, 0.6, by = 0.3)", # c(3,1)
  "runif(0, 0.6)" # c(3,2)
)
sensitivity_analysis_parallel_result <-
  sa(
    m1 = 1,
    x = data$covariates,
    y = data$y,
    w = data$w,
    prior_c_function = c_grid,
    n_cores = 1,
    estimand = "ATE",
  )
```

`summary.CIMTx_ATE_posterior` *Summarize a CIMTx_ATE_posterior object*

Description

Summarize a CIMTx_ATE_posterior object

Usage

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

Arguments

object	a CIMTx_ATE_posterior object obtained with ce_estimate function.
...	further arguments passed to or from other methods.

Value

a list with w*(w-1)/2 elements for ATE effect. Each element of the list contains the estimation, standard error, lower and upper 95% CI for RD/RR/OR.

References

Hadley Wickham (2019). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.4.0. URL:https://CRAN.R-project.org/package=stringr

Examples

```
library(CIMTx)
lp_w_all <-
  c(
    ".4*x1 + .1*x2  - .1*x4 + .1*x5", # w = 1
    ".2 * x1 + .2 * x2  - .2 * x4 - .3 * x5"
  ) # w = 2
nlp_w_all <-
  c(
    "-.5*x1*x4  - .1*x2*x5", # w = 1
    "-.3*x1*x4 + .2*x2*x5"
  ) # w = 2
lp_y_all <- rep(".2*x1 + .3*x2 - .1*x3 - .1*x4 - .2*x5", 3)
nlp_y_all <- rep(".7*x1*x1  - .1*x2*x3", 3)
X_all <- c(
  "rnorm(0, 0.5)", # x1
  "rbeta(2, .4)", # x2
  "runif(0, 0.5)", # x3
  "rweibull(1,2)", # x4
  "rbinom(1, .4)" # x5
)

set.seed(111111)
data <- data_sim(
  sample_size = 300,
  n_trt = 3,
  x = X_all,
  lp_y = lp_y_all,
  nlp_y = nlp_y_all,
  align = FALSE,
  lp_w = lp_w_all,
  nlp_w = nlp_w_all,
  tau = c(-1.5, 0, 1.5),
  delta = c(0.5, 0.5),
  psi = 1
)
```

```
ce_estimate_ra_ate_result <- ce_estimate(
  y = data$y, x = data$covariates,
  w = data$w, ndpost = 10, method = "RA", estimand = "ATE"
)
summary(ce_estimate_ra_ate_result)
```

`summary.CIMTx_ATE_sa` *Summarize a CIMTx_ATE_sa object*

Description

Summarize a CIMTx_ATE_sa object

Usage

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

Arguments

`object`	a `CIMTx_ATE_sa` object obtained with `sa` function.
`...`	further arguments passed to or from other methods.

Value

a data frame containing the estimation, standard error, lower and upper 95% CI for the causal estimand in terms of RD.

References

Hadley Wickham (2019). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.4.0. URL:https://CRAN.R-project.org/package=stringr

Examples

```
lp_w_all <-
  c(
    ".4*x1 + .1*x2  - 1.1*x4 + 1.1*x5", # w = 1
    ".2 * x1 + .2 * x2  - 1.2 * x4 - 1.3 * x5"
  ) # w = 2
nlp_w_all <-
  c(
    "-.5*x1*x4  - .1*x2*x5", # w = 1
    "-.3*x1*x4 + .2*x2*x5"
  ) # w = 2
lp_y_all <- rep(".2*x1 + .3*x2 - .1*x3 - 1.1*x4 - 1.2*x5", 3)
nlp_y_all <- rep(".7*x1*x1  - .1*x2*x3", 3)
X_all <- c(
```

```
  "rnorm(0, 0.5)", # x1
  "rbeta(2, .4)", # x2
  "runif(0, 0.5)", # x3
  "rweibull(1,2)", # x4
  "rbinom(1, .4)" # x5
)
set.seed(1111)
data <- data_sim(
  sample_size = 100,
  n_trt = 3,
  x = X_all,
  lp_y = lp_y_all,
  nlp_y = nlp_y_all,
  align = FALSE,
  lp_w = lp_w_all,
  nlp_w = nlp_w_all,
  tau = c(0.5, -0.5, 0.5),
  delta = c(0.5, 0.5),
  psi = 2
)
c_grid <- c(
  "runif(-0.6, 0)", # c(1,2)
  "runif(0, 0.6)", # c(2,1)
  "runif(-0.6, 0)", # c(2,3)
  "seq(-0.6, 0, by = 0.3)", # c(1,3)
  "seq(0, 0.6, by = 0.3)", # c(3,1)
  "runif(0, 0.6)" # c(3,2)
)
sensitivity_analysis_parallel_ATE_result <-
  sa(
    m1 = 1,
    x = data$covariates,
    y = data$y,
    w = data$w,
    prior_c_function = c_grid,
    nCores = 1,
    estimand = "ATE",
  )
summary(sensitivity_analysis_parallel_ATE_result)
```

`summary.CIMTx_ATT_posterior` *Summarize a CIMTx_ATT_posterior object*

Description

Summarize a CIMTx_ATT_posterior object

Usage

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

Arguments

object	a CIMTx_ATT_posterior object obtained with ce_estimate function.
...	further arguments passed to or from other methods.

Value

a list with w-1 elements for ATT effect. Each element of the list contains the estimation, standard error, lower and upper 95% CI for RD/RR/OR.

References

Hadley Wickham (2019). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.4.0. URL:https://CRAN.R-project.org/package=stringr

Examples

```
lp_w_all <-
  c(
    ".4*x1 + .1*x2  - .1*x4 + .1*x5", # w = 1
    ".2 * x1 + .2 * x2  - .2 * x4 - .3 * x5"
  ) # w = 2
nlp_w_all <-
  c(
    "-.5*x1*x4  - .1*x2*x5", # w = 1
    "-.3*x1*x4 + .2*x2*x5"
  ) # w = 2
lp_y_all <- rep(".2*x1 + .3*x2 - .1*x3 - .1*x4 - .2*x5", 3)
nlp_y_all <- rep(".7*x1*x1  - .1*x2*x3", 3)
X_all <- c(
  "rnorm(0, 0.5)", # x1
  "rbeta(2, .4)", # x2
  "runif(0, 0.5)", # x3
  "rweibull(1,2)", # x4
  "rbinom(1, .4)" # x5
)

set.seed(111111)
data <- data_sim(
  sample_size = 300,
  n_trt = 3,
  x = X_all,
  lp_y = lp_y_all,
  nlp_y = nlp_y_all,
  align = FALSE,
  lp_w = lp_w_all,
  nlp_w = nlp_w_all,
```

```
  tau = c(-1.5, 0, 1.5),
  delta = c(0.5, 0.5),
  psi = 1
)
ce_estimate_ra_att_result <- ce_estimate(
  y = data$y, x = data$covariates,
  w = data$w, reference_trt = 1, ndpost = 10, method = "RA", estimand = "ATT"
)
summary(ce_estimate_ra_att_result)
```

`summary.CIMTx_ATT_sa` *Summarize a CIMTx_ATT_sa object*

Description

Summarize a CIMTx_ATT_sa object

Usage

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

Arguments

`object`	a `CIMTx_ATT_sa` object obtained with `sa` function.
`...`	further arguments passed to or from other methods.

Value

a data frame containing the estimation, standard error, lower and upper 95% CI for the causal estimand in terms of RD.

References

Hadley Wickham (2019). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.4.0. URL:https://CRAN.R-project.org/package=stringr

Examples

```
lp_w_all <-
  c(
    ".4*x1 + .1*x2  - 1.1*x4 + 1.1*x5", # w = 1
    ".2 * x1 + .2 * x2  - 1.2 * x4 - 1.3 * x5"
  ) # w = 2
nlp_w_all <-
  c(
    "-.5*x1*x4  - .1*x2*x5", # w = 1
    "-.3*x1*x4 + .2*x2*x5"
```

```
  ) # w = 2
lp_y_all <- rep(".2*x1 + .3*x2 - .1*x3 - 1.1*x4 - 1.2*x5", 3)
nlp_y_all <- rep(".7*x1*x1  - .1*x2*x3", 3)
X_all <- c(
  "rnorm(0, 0.5)", # x1
  "rbeta(2, .4)", # x2
  "runif(0, 0.5)", # x3
  "rweibull(1,2)", # x4
  "rbinom(1, .4)" # x5
)
set.seed(1111)
data <- data_sim(
  sample_size = 100,
  n_trt = 3,
  x = X_all,
  lp_y = lp_y_all,
  nlp_y = nlp_y_all,
  align = FALSE,
  lp_w = lp_w_all,
  nlp_w = nlp_w_all,
  tau = c(0.5, -0.5, 0.5),
  delta = c(0.5, 0.5),
  psi = 2
)
c_grid <- c(
  "runif(-0.6, 0)", # c(1,2)
  "runif(0, 0.6)", # c(2,1)
  "runif(-0.6, 0)", # c(2,3)
  "seq(-0.6, 0, by = 0.3)", # c(1,3)
  "seq(0, 0.6, by = 0.3)", # c(3,1)
  "runif(0, 0.6)" # c(3,2)
)
sensitivity_analysis_parallel_ATT_result <-
  sa(
    m1 = 1,
    x = data$covariates,
    y = data$y,
    w = data$w,
    prior_c_function = c_grid,
    nCores = 1,
    estimand = "ATE",
  )
summary(sensitivity_analysis_parallel_ATT_result)
```

`summary.CIMTx_IPTW` *Summarize a CIMTx_IPTW object*

Description

Summarize a CIMTx_IPTW object

Usage

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

Arguments

object	a CIMTx_IPTW object
...	further arguments passed to or from other methods.

Value

a data frame with ATT/ATE effect estimates in terms of RD, RR and OR.

References

Hadley Wickham (2019). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.4.0. URL:https://CRAN.R-project.org/package=stringr

Examples

```
lp_w_all <-
  c(
    ".4*x1 + .1*x2  - .1*x4 + .1*x5", # w = 1
    ".2 * x1 + .2 * x2  - .2 * x4 - .3 * x5"
  ) # w = 2
nlp_w_all <-
  c(
    "-.5*x1*x4  - .1*x2*x5", # w = 1
    "-.3*x1*x4 + .2*x2*x5"
  ) # w = 2
lp_y_all <- rep(".2*x1 + .3*x2 - .1*x3 - .1*x4 - .2*x5", 3)
nlp_y_all <- rep(".7*x1*x1  - .1*x2*x3", 3)
X_all <- c(
  "rnorm(0, 0.5)", # x1
  "rbeta(2, .4)", # x2
  "runif(0, 0.5)", # x3
  "rweibull(1,2)", # x4
  "rbinom(1, .4)" # x5
)

set.seed(111111)
data <- data_sim(
  sample_size = 300,
  n_trt = 3,
  x = X_all,
  lp_y = lp_y_all,
  nlp_y = nlp_y_all,
  align = FALSE,
  lp_w = lp_w_all,
  nlp_w = nlp_w_all,
  tau = c(-1.5, 0, 1.5),
```

```
  delta = c(0.5, 0.5),
  psi = 1
)
iptw_multi_res <- ce_estimate(
  y = data$y, x = data$covariates,
  w = data$w, method = "IPTW-Multinomial", estimand = "ATE"
)
summary(iptw_multi_res)
```

---

summary.CIMTx_nonIPTW_once *Summarize a CIMTx_nonIPTW_once object*

Description

Summarize a CIMTx_nonIPTW_once object

Usage

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

Arguments

object	a CIMTx_nonIPTW_once object
...	further arguments passed to or from other methods.

Value

a data frame with ATT/ATE effect estimates in terms of RD, RR and OR.

References

Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2021). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.7. URL: https://CRAN.R-project.org/package=dplyr

Hadley Wickham (2019). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.4.0. URL:https://CRAN.R-project.org/package=stringr

Examples

```
lp_w_all <-
  c(
    ".4*x1 + .1*x2  - .1*x4 + .1*x5", # w = 1
    ".2 * x1 + .2 * x2  - .2 * x4 - .3 * x5"
  ) # w = 2
nlp_w_all <-
```

```
  c(
    "-.5*x1*x4  - .1*x2*x5", # w = 1
    "-.3*x1*x4 + .2*x2*x5"
  ) # w = 2
lp_y_all <- rep(".2*x1 + .3*x2 - .1*x3 - .1*x4 - .2*x5", 3)
nlp_y_all <- rep(".7*x1*x1  - .1*x2*x3", 3)
X_all <- c(
  "rnorm(0, 0.5)", # x1
  "rbeta(2, .4)", # x2
  "runif(0, 0.5)", # x3
  "rweibull(1,2)", # x4
  "rbinom(1, .4)" # x5
)

set.seed(111111)
data <- data_sim(
  sample_size = 300,
  n_trt = 3,
  x = X_all,
  lp_y = lp_y_all,
  nlp_y = nlp_y_all,
  align = FALSE,
  lp_w = lp_w_all,
  nlp_w = nlp_w_all,
  tau = c(-1.5, 0, 1.5),
  delta = c(0.5, 0.5),
  psi = 1
)
iptw_tmle_res <- ce_estimate(
  y = data$y, x = data$covariates,
  w = data$w, method = "TMLE", estimand = "ATE",
  sl_library = c("SL.glm", "SL.glmnet")
)
summary(iptw_tmle_res)
```

`true_c_fun_cal` *Calculate the true c functions with 3 treatments and a binary predictor*

Description

This function calculates the true confounding functions with 3 treatments and a binary predictor for simulated data.

Usage

```
true_c_fun_cal(x, w)
```

Arguments

x	A matrix with one column for the binary predictor with values 0 and 1
w	A treatment indicator

Value

A matrix with 2 rows and 6 columns

Examples

```
set.seed(111)
data_SA <- data_sim(
  sample_size = 100,
  n_trt = 3,
  x = c(
    "rbinom(1, .5)", # x1:measured confounder
    "rbinom(1, .4)"
  ), # x2:unmeasured confounder
  lp_y = rep(".2*x1+2.3*x2", 3), # parallel response surfaces
  nlp_y = NULL,
  align = FALSE, # w model is not the same as the y model
  lp_w = c(
    "0.2 * x1 + 2.4 * x2", # w = 1
    "-0.3 * x1 - 2.8 * x2"
  ),
  nlp_w = NULL,
  tau = c(-2, 0, 2),
  delta = c(0, 0),
  psi = 1
)
x1 <- data_SA$covariates[, 1, drop = FALSE]
w <- data_SA$w
Y1 <- data_SA$Y_true[, 1]
Y2 <- data_SA$Y_true[, 2]
Y3 <- data_SA$Y_true[, 3]
true_c_fun <- true_c_fun_cal(x = x1, w = w)
```

---

trunc_fun *Trimming*

Description

The function trims the weights for IPTW methods.

Usage

```
trunc_fun(x, trim_perc = 0.05)
```

Arguments

x	A numeric vector
trim_perc	A numeric vector with length 2 indicating trimming percentile.

Value

A numeric vector

Index