Package ‘regulaTEr’

August 14, 2025

Title regulaTEr

Version 0.0

Description What the package does (one paragraph).

License `use_mit_license()`, `use_gpl3_license()` or friends to pick a license

Encoding UTF-8

Imports glmnet,
ggplot2

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0)

URL https://github.com/lsun20/RegulATE

Contents

absnorm_q *Quantile of folded standard normal distribution*

Description

Computes the quantile of the folded standard normal distribution, i.e., the distribution of the absolute value of a normally distributed variable with mean x and standard deviation 1. If $x > 3$, the function uses a normal approximation; otherwise, it uses the square root of a noncentral chi-squared quantile.

Usage

```
absnorm_q(x, q)
```

Arguments

`x` A numeric vector; the mean of the standard normal distribution.

`q` A numeric vector; the desired quantile level between 0 and 1.

Value

A numeric vector; the q-th quantile of $|Z|$, where $Z \sim N(x, 1)$.

calculate_lambda_range *Calculate an appropriate range for finding the optimal lambda value.*

Description

Determines a feasible search interval for the regularization parameter λ used in penalized propensity score regression. The minimum value of this interval is always set to zero. The maximum is initialized using the largest λ value returned by `glmnet` because it recovers the fully regularized short regression coefficient estimates. The maximum λ is then adaptively increased by 2 until the ℓ_2-distance between the weight vectors of this penalized propensity score regression and that of the fully penalized propensity score regression falls below a pre-specified tolerance.

Usage

```
calculate_lambda_range(
  res_prop,
  B,
  resid_Z_w_Z_uc,
  x,
  short_wghts,
  scaling = 2
)
```

Arguments

`res_prop` A numeric vector; residuals from projecting `x` onto `Z_uc`.

`B` A numeric matrix; the covariance matrix of the demeaned `Z_uc`.

`resid_Z_w_Z_uc` A numeric matrix; residualized `Z_w` after projection onto `Z_uc`.

`x` A numeric vector; realizations corresponding to the treatment variable.

`short_wghts` A numeric vector; weights used in the short regression specification.

`scaling` A numeric scalar; a scaling factor for the maximum λ value. Defaults to 2.

Value

A numeric vector; containing the minimum and maximum value of the lambda range.

`cal_C` *Estimate* `C`.

Description

Estimate `C`, the heterogeneity bound, from the data. If the long regression includes covariates with `NA` coefficients, use the trimmed data instead.

Usage

```
cal_C(Y, x, Z_uc, Z_w, B, trimmed_data, na_covariates)
```

Arguments

`Y` A numeric vector; realizations of the outcome variable.

`x` A numeric vector; realizations of the treatment variable.

`Z_uc` A numeric matrix; realizations of the control covariates.

`Z_w` A numeric matrix; realizations of the interacted covariates.

`B` A numeric matrix; covariance matrix of the demeaned `Z_uc`.

`trimmed_data` A list; output from `trim_data()`, used if the long regression has `NA` coefficients.

`na_covariates` A character vector; names of interaction covariates with `NA` coefficients with the `Z_w` prefix removed obtained from running `find_long_NAs()`.

Value

A list containing the square of the estimated `C` and its associated standard error, and CI.

check_valid_inputs *Verify user inputs to regulaTE() are valid.*

Description

Verify user inputs to regulaTE() are valid.

Usage

```
check_valid_inputs(
  outcome,
  treatment,
  covariates,
  parameter,
  C,
  se,
  cluster,
  sig_level,
  df_corr,
  digits,
  confounders,
  trimmed_data
)
```

Arguments

outcome A numeric vector; realizations of the outcome variable.

treatment A numeric vector; realizations of the treatment variable.

covariates A numeric matrix; realizations of the control covariates.

parameter A string; the parameter of interest, which must be "ATE", "ATT", or "ATU". Defaults to "ATE".

C A numeric scalar or vector; contains non-negative values specifying the heterogeneity bound or NA if one wishes the program to estimate C using the data.

se A string; either "het", or "hom". "hom" computes homoskedastic standard errors. "het" computes heteroskedastic-robust standard errors. If "het" is chosen and cluster is specified, then it computes clustered heteroskedastic-robust standard errors.

cluster A vector; cluster identifier for computing clustered standard errors. Defaults to NULL. If NULL, non-clustered standard errors are used.

sig_level A numeric vector; significance level for confidence intervals. Defaults to 0.05.

df_corr indicating whether to apply degrees-of-freedom correction. Defaults to TRUE.

digits An integer vector; number of decimal places to use when formmatting numeric values. Defaults to 4.

confounders A numeric matrix; realizations of the confounders. Defaults to NULL. If NULL, the function will use the covariates as confounders.

trimmed_data A list; contains the matrices/vectors for the outcome (Y), treatment indicator (x), and interaction variables (Z_w), plus either covariates (Z_uc) or confounders (Z_conf). Used when trimming is necessary due to lack of overlap and the user wants to specify the trimmed data directly. Defaults to NULL, in which case the function trims the data automatically.

Value

None if inputs are valid; otherwise, the function stops the program.

`check_valid_plot_inputs` *Checks if inputs into plot_regulaTE() are valid.*

Description

Checks if inputs into plot_regulaTE() are valid.

Usage

```
check_valid_plot_inputs(estimates, y_axis_title, title = NULL)
```

Arguments

`estimates` A data frame; `estimates_df` from regulaTE().

`y_axis_title` A character string; the title for the y-axis. Defaults to "Outcome."

`title` A character string; the tile for the graph. Defaults to "Estimates under Heterogeneity Bound for "

Value

None if inputs are valid; otherwise, the function stops the program.

`clean_estimates_table` *Clean parameter estimates table for formatted output.*

Description

Produces a cleaned version of the estimates table from `opt_het_bound()`, formatted for display via `print_pretty_table()`.

Usage

```
clean_estimates_table(table, parameter, digits, sig_level)
```

Arguments

`table` A data frame; contains the point estimates, standard errors, and confidence intervals corresponding to the parameter estimates from `opt_het_bound()`.

`parameter` A character vector; the parameter of interest, which must be one of `"ATE"`, `"ATT"`, or `"ATU"`.

`digits` An integer vector; number of decimal places to use when formmatting numeric values.

`sig_level` A numeric vector; significance level for confidence intervals.

Value

A data frame; contains the point estimates, standard errors, and confidence intervals corresponding to the parameter estimates from `opt_het_bound()` formalized appropriately for console output.

console_output *Console output for regulaTE().*

Description

Prints formatted console output for results from `opt_het_bound()`.

Usage

```
console_output(
  output,
  estimates_table,
  parameter,
  digits,
  sig_level,
  C,
  multi = FALSE
)
```

Arguments

`output`	A list; output from `opt_het_bound()`
`estimates_table`	A data frame; produced by `produce_estimates_table()`.
`parameter`	A character string; the parameter of interest, which must be one of `"ATE"`, `"ATT"`, or `"ATU"`.
`digits`	An numeric vector; number of decimal places to display in the output.
`sig_level`	A numeric vector; significance level for confidence intervals.
`C`	A numeric vector; non-negative values specifying the heterogeneity bound or `NA` if one wishes the program to estimate `C` using the data.
`multi`	A logical vector; whether the output is based on multiple `C` values. Defaults to `FALSE`.

Value

None. Only prints to console.

covariate_comps *Preliminary covariate computations*

Description

Performs preliminary computations only dependent on the covariates. Computes demeaned covariates and interaction terms baese on the specified treatment effect parameter.

Usage

```
covariate_comps(x, Z_uc, parameter = "ATE")
```

Arguments

x	A numeric vector; realizations of the treatment variable.
Z_uc	A numeric matrix; realizations of the control covariates.
parameter	A character string; Specifies which treatment effect ("ATE", "ATU", or "ATT") to compute. Defaults to "ATE".

Value

A list containing the following components:

- `parameter`: A character string; the specified treatment effect ("ATE", "ATT", or "ATU").
- `n`: An integer; number of observations (rows).
- `x`: A numeric vector;realizations of the treatment variable.
- `Z_uc`: A numeric matrix; realizations of the control covariates.
- `Z_tilde`: A numeric matrix; the demeaned version of `Z_uc` according to the specified `parameter`.
- `Z_w`: A numeric matrix; realizations cof the interacted covariates.

cr_se_loop *Cluster-robust standard error computation*

Description

Computes cluster-robust, heteroskedasticity-robust standard errors for a linear combination of residuals and weights, with optional finite-sample degrees-of-freedom correction. It is thus equivalent to the `HC1` estimator (default in Stata).

Usage

```
cr_se_loop(wghts, residuals, cluster, num_covar, df_corr = TRUE)
```

Arguments

`wghts`	A numeric vector; weights applied to the residuals.
`residuals`	A numeric vector; residuals.
`cluster`	A vector; cluster identifiers for each obsevation.
`num_covar`	An integer vector; the number of covariates in the regression model.
`df_corr`	A logical vector; if `TRUE`, applies a finite-sample correction to the standard error calculation.

Value

A numeric scalar corresponding to the cluster-robust standard error.

`determine_residuals` *Long or ridge regression residuals*

Description

Returns residuals, either from the long regression or ridge long regression, depending on whether the long regression contains any `NA` coefficients.

Usage

```
determine_residuals(long_NAs, long_reg, ridgereg)
```

Arguments

`long_NAs`	A character vector; names of interaction covariates with `NA` coefficients, with the `Z_w` prefix removed. Possibly empty.
`long_reg`	An `lm` object; the long regression object, `lm(Y ~ -1 + x + Z_uc + Z_w)`.
`ridgereg`	A `glmnet` object; the ridge regression object obtained from running `run_long_ridge()`.

Value

A list containing the following components:

- `residuals`: A numeric vector; residuals, either from the long regression or ridge long regression.
- `df`: A numeric vector; degrees of freedom corresponding to the long regression or ridge long regression.

find_long_NAs — NA *coefficients in long regression detector*

Description

Returns the names of the NA coefficients in the interaction covariates of a given long regression: these are the coefficients contributing to lack of overlap and hence a long regression with NA coefficients.

Usage

```
find_long_NAs(long_reg)
```

Arguments

long_reg	An lm object; the long regression object, lm(Y ~ -1 + x + Z_conf + Z_w).

Value

A character vector; containing the names of interaction covariates with NA coefficients with the Z_w prefix removed. If there are no such covariates, returns an empty character vector.

generate_estimates_table — *Generate a table of estimates for each regression specifcation*

Description

Generate a table of estimates for each regression specifcation

Usage

```
generate_estimates_table(
  opt_results,
  reg_estimates,
  sigma,
  alpha,
  zalpha,
  cva,
  max_bias_short,
  max_bias_trim,
  cluster,
  df_corr
)
```

Arguments

`opt_results` A list; output from running `opt_lam()`.

`reg_estimates` A list; components obtained from running `reg_estimates()`.

`sigma` A numeric vector; residual standard deviation used to compute robust SEs when clustering is not specified.

`alpha` A numeric vector; between `0` and `1`; significance level for CIs.

`zalpha` A numeric vector; standard normal quantile for two-sided CI.

`cva` A numeric vector; quantile from folded normal distribution used for bias correction of the short regression.

`max_bias_short` A numeric vector; the maximum bias of the short regression.

`max_bias_trim` A numeric vector; the maximum bias of the trimmed long regression

`cluster` A vector; optional cluster identifiers for each obsevation. If `NULL`, standard errors are not clustered.

`df_corr` A logical vector; indicating whether to apply degrees-of-freedom correction. Defaults to `TRUE`.

Value

A numeric matrix; each row corresponds to a regression specification and includes point estimates (PE), standard errors (SE), confidence intervals (CI LB and CI UB), and maximum bias, and critical value.

`opt_het_bound` *Inference on treatment effects using a bound on heterogeneity*

Description

Estimates the treatment effects (ATE, ATT, or ATU) using four different regressions: Opt, Short, Short BC, (Trimmed) Long.

Usage

```
opt_het_bound(
  Y,
  reg_estimates,
  cluster,
  C,
  alpha = 0.05,
  df_corr = TRUE,
  scaling = 2
)
```

Arguments

`Y`	A numeric vector; outcome variable.
`reg_estimates`	A list; output from `reg_estimates()`.
`cluster`	A vector; cluster identifier for computing clustered standard errors. If `NULL`, non-clustered standard errors are used.
`C`	A numeric vector; heterogeneity bound. If `NULL`, estimated using `cal_C()`.
`alpha`	A numeric vector; significance level for confidence intervals. Defaults to 0.05.
`df_corr`	A logical; indicating whether to apply degrees-of-freedom correction. Defaults to `TRUE`.
`scaling`	A numeric scalar; a scaling factor for the maximum λ value. Defaults to 2.

Value

A list with the following components:

- `estimates_table`: A numeric matrix; estimates for Opt, Short, Short BC, and (Trimmed) Long specs produced by `generate_estimates_table()`.
- `c_table`: A numeric matrix or `NA`; If `C` is estimated, a matrix containing PE, SE, and CI for `C` obtained from `generate_estimates_table()`; otherwise `NA`.
- `trimming_table`: A numeric matrix or `NA`; If trimming of the long regression is required, a matrix containing details of the trimming process obtained from `generate_estimates_table()`; otherwise `NA`.
- `lambda`: A numeric vector; value of the regularization parameter that minimizes CI length.
- `C_orig`: A numeric vector or `NULL`; the user-supplied value of `C` or `NULL` if estimated using `cal_C()`.
- `samp_size`: A numeric vector; number of observations in the data.
- `num_covar_short`: A numeric vector; number of covariates used in the short regression.
- `num_covar_long`: A numeric vector; number of covariates used in the long regression.
- `lind_wt`: A numeric vector; the Lindeberg weight.
- `max_bias_short`: A numeric vector; the maximum bias of the short regression.

`plot_regulaTE` *Plots multiple estimates from regulaTE() with varying* `C` *values.*

Description

Plots multiple estimates from regulaTE() with varying `C` values.

Usage

```
plot_regulaTE(
  estimates,
  y_axis_title = "Outcome",
  title = NULL,
  show_long = FALSE
)
```

Arguments

`estimates`	A data frame; `estimates_df` from `regulaTE()`.
`y_axis_title`	A character string; the title for the y-axis. Defaults to "Outcome."
`title`	A character string; the title for the plot. Defaults to "Estimates under Heterogeneity Bound for "
`show_long`	A logical; whether to show the long regression results in the plot. Defaults to FALSE.

Value

Note. Only prints a plot.

`print_pretty_table` *Prints a well-formatted table to the console*

Description

Prints a well-formatted table to the console for a data frame of parameter or `C` estimates.

Usage

```
print_pretty_table(df, digits, pad = 2)
```

Arguments

`df`	A data frame; contains parameter of `C` estimates from `produce_estimates_table()` or `produce_c_table()`.
`digits`	An integer vector; number of decimal places to use when formmatting numeric values.
`pad`	An integer vector; number of spaces between the columns. Defaults to 2.

Value

None. Only prints to console.

`produce_c_table` *Produces `C` estimates table for formatted output.*

Description

Produces a cleaned data frame containing the estimated C^2, standard error, and confidence interval, formatted for display via `print_pretty_table()`.

Usage

```
produce_c_table(table, digits, sig_level)
```

Arguments

table A data frame; contains the estimates.

digits An integer vector; number of decimal places to use when formmatting numeric values.

sig_level A numeric vector; significance level for confidence intervals.

Value

A data frame; contains the point estimate, standard error, and confidence interval corresponding to the estimated $\vec{C}$ from opt_het_bound() formalized appropriately for console output.

regulaTE *Function performing inference under a heterogeneity bound that end users call.*

Description

Function performing inference under a heterogeneity bound that end users call.

Usage

```
regulaTE(
  outcome,
  treatment,
  covariates,
  confounders = NULL,
  parameter = "ATE",
  C = NA,
  se = "hom",
  cluster = NULL,
  sig_level = 0.95,
  df_corr = TRUE,
  digits = 4,
  trimmed_data = NULL,
  scaling = 2
)
```

Arguments

outcome A numeric vector; realizations of the outcome variable.

treatment A numeric vector; realizations of the treatment variable.

covariates A numeric matrix; realizations of the control covariates.

confounders A numeric matrix; realizations of the confounders. Defaults to NULL. If NULL, the function will use the covariates as confounders.

parameter A string; the parameter of interest, which must be "ATE", "ATT", or "ATU". Defaults to "ATE".

C A numeric scalar or vector; contains non-negative values specifying the heterogeneity bound or NA if one wishes the program to estimate C using the data.

`se` A string; either `"het"`, or `"hom"`. `"hom"` computes homoskedastic standard errors. `"het"` computes heteroskedastic-robust standard errors. If `"het"` is chosen and `cluster` is specified, then it computes clustered heteroskedastic-robust standard errors.

`cluster` A vector; cluster identifier for computing clustered standard errors. Defaults to `NULL`. If `NULL`, non-clustered standard errors are used.

`sig_level` A numeric vector; significance level for confidence intervals. Defaults to 0.05.

`df_corr` indicating whether to apply degrees-of-freedom correction. Defaults to `TRUE`.

`digits` An integer vector; number of decimal places to use when formmatting numeric values. Defaults to 4.

`trimmed_data` A list; contains the matrices/vectors for the outcome (`Y`), treatment indicator (`x`), and interaction variables (`Z_w`), plus either covariates (`Z_uc`) or confounders (`Z_conf`). Used when trimming is necessary due to lack of overlap and the user wants to specify the trimmed data directly. Defaults to `NULL`, in which case the function trims the data automatically.

`scaling` A numeric scalar; a scaling factor for the maximum λ value. Defaults to 2.

Value

A list of two data frames:

- `summary`: A data frame; contains the sample size, number of covariates in the short and long regressions, and the Lindeberg weight.
- `estimates`: A data frame; contains point estimates, standard errors, and confidence intervals for each specification (Opt, Short, Short BC, and Long).

`reg_estimates` *Short, long, and ridge regressions*

Description

Estimates long and short regressions, as well as a trimmed long and ridge regression (via `run_long_ridge()`) if `NA` coefficients exist.

Usage

```
reg_estimates(Y, x, covariate_comps, Z_conf = NULL, trimmed_data = NULL)
```

Arguments

`Y` A numeric vector; realizations of the outcome variable.

`x` A numeric vector; realizations of the treatment variable.

`covariate_comps` A list; covariate-related components returned by `covariate_comps()`.

`Z_conf` A numeric matrix; realizations of the confounder covariates. Defaults to `Z_uc` if `NULL`.

`trimmed_data` A list; contains the matrices/vectors for the outcome (`Y`), treatment indicator (`x`), and interaction variables (`Z_w`), plus either covariates (`Z_uc`) or confounders (`Z_conf`). Used when trimming is necessary due to lack of overlap and the user wants to specify the trimmed data directly. Defaults to `NULL`, in which case the function trims the data automatically.

Value

A list with the following components:

- `multi_results`: A list; contains (if applicable):
  - `ridgereg`: A `glmnet` object; the ridge regression object (or `NULL` if unused).
  - `trim_reg`: An `lm` object; the trimmed long regression object.
  - `res_long_prop_trim`: A numeric vector; residuals from projecting `x` on trimmed covariates.
  - `long_wghts_trim`: A numeric vector; weights from trimmed long regression.
  - `long_wghts_trim_full`: A numeric vector; weights from trimmed long regression, but with zeros at trimmed rows.
  - `rows_to_delete`: An integer vector; trimmed row indices.
- `covariate_comps`: A list; the original, unchanged input to the function.
- `long_reg`: A `lm` object; `Y ~ - 1 + x + Z_conf + Z_w`.
- `long_NAs`: An integer vector; indices for coefficients dropped in the long regression.
- `num_covar_long`: An integer; number of covariates in the long regression.
- `num_covar_trim`: An integer; number of covariates in the trimmed regression.
- `shortreg`: An `lm` object; `Y ~ - 1 + x + Z_conf`.
- `residuals_df`: A list; obtained from `determine_residuals()` with:
  - `res`: A numeric vector; residual, either from the long or ridge long regression.
  - `df`: An integer; residual degrees of freedom
- `sigma`: A numeric vector; estimated residual standard deviation.
- `wghts`: A numeric matrix; the matrix cross-product of `short_wghts` and `Z_w`.
- `res_long_prop`: A numeric vector; residuals from projecting `x` onto `Z_conf` and `Z_w`.
- `long_wghts`: A numeric vector; weights used in the long regression specification.
- `num_covar_long`: An integer; number of covariates in the long regression.
- `res_prop`: A numeric vector; residuals from projecting `x` onto `Z_conf`.
- `short_wghts`: A numeric vector; weights used in the short regression specification.
- `short_prop`: An `lm` object; corresponds to the regression of `x` on `Z_conf`.
- `resid_Z_w_Z_uc`: A numeric matrix; residualized `Z_w` after projection onto `Z_conf`.
- `Z_w_resid_Z_w_Z_uc`: A numeric matrix; matrix cross-product of residualized `Z_w`.
- `Z_w_res_prop`: A numeric vector; matrix cross-product of residualized `Z_w` with `res_prop`.
- `B`: A numeric matrix; the covariance matrix of the demeaned `Z_uc`.
- `inv_B`: A numeric matrix; the inverse of `B`.

run_long_ridge *Ridge long regression*

Description

Runs a ridge long regression with a penalty determined by 10-fold cross-validation and computes the residuals and effective degrees of freedom. Used when the long regression has `NA` coefficients.

Usage

```
run_long_ridge(Y, x, Z_conf, Z_w, B)
```

Arguments

`Y`	A numeric vector; realizations of the outcome variable.
`x`	A numeric vector; realizations of the treatment variable.
`Z_conf`	A numeric matrix; realizations of the confounder covariates.
`Z_w`	A numeric matrix; realizations cof the interacted covariates.
`B`	A numeric matrix; covariance matrix of the demeaned `Z_uc`.

Value

A list containing the following components:

- `ridge`: An `glmnet` object; fitted ridge regression model, an object of class `glmnet`.
- `residuals`: A numeric vector; residuals.
- `df`: A numeric vector; the effective degrees of freedom for the ridge regression model, computed as `n - sum(d^2 / (d^2 + lambda))`, where `d` are the singular values of the design matrix `X` and `lambda` is the optimal penalty determined by cross-validation.

sqrt_mat *Matrix square root*

Description

Computes the (approximate) principal square root B of a matrix A, so that $BB \approx A$, via Schur decomposition.

Usage

```
sqrt_mat(A)
```

Arguments

`A`	A numeric matrix.

Value

A numeric matrix; representing the (approximate) principal square root of `A`.

`trim_data` *Trim data associated with limited overlap.*

Description

Trims rows and columns that cause an undefined long regression. This function should be used only when `Z_conf` = `Z_uc` (i.e., no confounders).

Usage

```
trim_data(data, long_NAs, parameter)
```

Arguments

`data`	A matrix; `cbind(Y, x, Z_uc)`, where `Y`, `x`, and `Z_uc` correspond to realizations of the outcome variable, the treatment variable, and the covariate variables, respectively.
`long_NAs`	A character vector; names of interaction covariates with `NA` coefficients, with the `Z_w` prefix removed. Possibly empty.
`parameter`	A string; the parameter of interest, which must be `"ATE"`, `"ATT"`, or `"ATU"`. Defaults to `"ATE"`.

Value

A list containing the following components:

- `Y`: A numeric vector; realizations of the outcome variable after trimming.
- `x`: A numeric vector; realizations of the treatment variable after trimming.
- `Z_uc`: A numeric matrix; realizations of the covariates after trimming.
- `Z_w`: A numeric matrix; realizations of the interacted covariates after trimming.
- `Z_tilde`: A numeric matrix; demeaned version of `Z_uc`, based on `parameter`.
- `B_trim`: A numeric matrix; covariance matrix of the demeaned `Z_uc`.
- `trimming_table`: A numeric matrix; 2-row summary of trimming across rows and columns.
- `trimmed_row_indices`: A numeric vector; indices of rows removed due to limited overlap.

Here "trimmed" means trimmed according to the procedures in `trim_na_rows()` and `trim_na_cols()`.

trim_na_cols *Removes NA-causing columns (columns associated limited overlap) in dataset.*

Description

Helper function for `trim_data()`. Removes ("trims") all columns whose names are included in `na_covariates`.

Usage

```
trim_na_cols(data, na_covariates)
```

Arguments

`data`	A matrix; `cbind(Y, x, Z_uc)`, where `Y`, `x`, and `Z_uc` correspond to realizations of the outcome variable, the treatment variable, and the covariate variables, respectively.
`na_covariates`	A character vector; contains the names of interaction covariates with `NA` coefficients with the `Z_w` prefix removed obtained from running `find_long_NAs()`.

Value

A list containing the following components:

- `data`: The dataset with NA columns removed.
- `cols_before`: A numeric scalar; equal to the number of columns in the data before trimming.
- `cols_after`: A numeric saclar; equal to the number of columns in the data after trimming.
- `cols_trimmed`: A numeric scalar; equal to the number of columns trimmed.
- `cols_trimmed_percentage`: A numeric scalar; equal to the percentage of how many columns were trimmed.

trim_na_rows *Helper function for* `trim_data()`*. Removes ("trims") all rows that have a value of 1 for any of the covariates passed through in* `na_covariates`*.*

Description

Helper function for `trim_data()`. Removes ("trims") all rows that have a value of 1 for any of the covariates passed through in `na_covariates`.

Usage

```
trim_na_rows(data, na_covariates)
```

Arguments

`data` A matrix; `cbind(Y, x, Z_uc)`, where `Y`, `x`, and `Z_uc` correspond to realizations of the outcome variable, the treatment variable, and the covariate variables, respectively.

`na_covariates` A character vector; contains the names of interaction covariates with `NA` coefficients with the `Z_w` prefix removed obtained from running `find_long_NAs()`.

Value

A list containing the following components:

- `data`: The dataset with NA rows removed.
- `deleted_rows`: A numeric vector; contains the indices of the rows that were deleted.
- `rows_before`: A numeric scalar; equal to the number of rows in the data before trimming.
- `rows_after`: A numeric saclar; equal to the number of rows in the data after trimming.
- `rows_trimmed`: A numeric scalar; equal to the number of rows trimmed.
- `rows_trimmed_percentage`: A numeric scalar; equal to the percentage of how many rows were trimmed.

Index