Package ‘rar’

January 24, 2024

Title Risk-Adjusted Regression

Version 0.0.3

Description Perform risk-adjusted regression and sensitivity analysis as developed in ``Mitigating Omitted- and Included-Variable Bias in Estimates of Disparate Impact'' Jung et al. (2024) <arXiv:1809.05651>.

License MIT + file LICENSE

Encoding UTF-8

URL `https://rar.jgaeb.com`, `https://github.com/jgaeb/rar`

BugReports `https://github.com/jgaeb/rar/issues`

RoxygenNote 7.2.3

LinkingTo cpp11, testthat

Suggests broom, forcats, stringr, testthat (>= 3.0.0), xml2

Config/testthat/edition 3

Imports dplyr, glue, magrittr, purrr, rlang, tibble, tidyr, tidyselect, vctrs

NeedsCompilation yes

Author Johann Gaebler [aut, cre, cph] (<`https://orcid.org/0000-0003-3340-9542`>)

Maintainer Johann Gaebler <`me@jgaeb.com`>

Repository CRAN

Date/Publication 2024-01-24 17:30:05 UTC

R topics documented:

`sens` *Perform sensitivity analysis on a risk-adjusted regression*

Description

`sens()` performs sensitivity analysis on a risk-adjusted regression by computing the maximum and minimum regression coefficients consistent with the data and the analyst's prior knowledge, expressed through `epsilon`, the bound on the mean absolute difference between the true and estimated risks. It additionally can provide bootstrapped pointwise confidence intervals for the regression coefficients.

Usage

```
sens(
  df,
  group_col,
  obs_col,
  p_col,
  base_group,
  epsilon,
  lwr_col = NULL,
  upr_col = NULL,
  eta = 0.01,
  m = 101L,
  N = 0L,
  alpha = 0.05,
  chunk_size = 100L,
  n_threads = 1L
)
```

Arguments

`df` The data frame containing the data.

`group_col` The name of the column containing the group labels. This column should be a factor or coercible to a factor.

`obs_col` The name of the column containing whether or not the outcome was observed. This column should be a logical or coercible to a logical.

`p_col` The name of the column containing the estimated risks. These risks should be expressed on the probability scale, i.e., be between 0 and 1.

`base_group` The name of the base group. This group will be used as the reference group in the regression.

`epsilon` The bound on the mean absolute difference between the true and estimated risks.

`lwr_col` The name of the column containing the lower bounds on the true risk. (Defaults to 0 for all observations.)

`upr_col` The name of the column containing the upper bounds on the true risk. (Defaults to 1 for all observations.)

`eta` The step size for the grid search. Note that while steps are taken at the group level, the step size is expressed at the level of change in average risk *across the entire population*. In other words, smaller groups will have proportionally larger steps. (Defaults to 0.01.)

`m` The grid size for the maximization approximation. (Defaults to `101`.)

`N` The number of bootstrap resamples to use to compute pointwise confidence intervals. (Defaults to 0, which performs no bootstrap.)

`alpha` The confidence level for the pointwise confidence intervals. (Defaults to 0.05.)

`chunk_size` The number of repetitions to perform in each chunk when run in parallel. Larger chunk sizes make it less likely that separate threads will block on each other, but also make it more likely that the threads will finish at different times. (Defaults to 100.)

`n_threads` The number of threads to use when running in parallel. (Defaults to 1, i.e., serial execution.)

Value

A data frame containing the following columns:

- `epsilon`: Values of epsilon ranging from 0 to the input value of `epsilon` in `m` steps.
- `beta_min_{group}`: The minimum value of the regression coefficient for the group `group`. (Note that the base group is not included in this list.)
- `beta_max_{group}`: The maximum value of the regression coefficient for the group `group`. (Note that the base group is not included in this list.)
- (**If** `N > 0`) `beta_min_{group}_{alpha/2}`: The `alpha/2` quantile of the bootstrap distribution of the minimum value of the regression coefficient for group `group`. (Note that the base group is not included in this list.)
- (**If** `N > 0`) `beta_min_{group}_{1 - alpha/2}`: The `1 - alpha/2` quantile of the bootstrap distribution of the minimum value of the regression coefficient for group `group`. (Note that the base group is not included in this list.)
- (**If** `N > 0`) `beta_max_{group}_{alpha/2}`: The `alpha/2` quantile of the bootstrap distribution of the maximum value of the regression coefficient for group `group`. (Note that the base group is not included in this list.)
- (**If** `N > 0`) `beta_max_{group}_{1 - alpha/2}`: The `1 - alpha/2` quantile of the bootstrap distribution of the maximum value of the regression coefficient for group `group`. (Note that the base group is not included in this list.)

Details

The sensitivity analysis assumes that every group contains at least one observed and one unobserved individual, and that the estimated risks and upper and lower bounds are "sortable," i.e., that there exists a permutation of the rows such that the estimated risks and upper and lower bounds are all non-decreasing within each group and observation status. If these conditions are not met, the function will throw an error.

To ensure that these conditions continue to hold, the bootstrap resamples are stratified by group and observation status. As a result, in small samples, the confidence intervals may be slightly narrowed, since they do not account for uncertainty in the number of individuals in each group, and the number of observed and unobserved individuals within each group.

Examples

```
# Generate some data
set.seed(1)
df <- tibble::tibble(
  group = factor(
    sample(c("a", "b"), 1000, replace = TRUE),
    levels = c("a", "b")
  ),
  p = runif(1000)^2,
  frisked = runif(1000) < p + 0.1 * (group != "a")
)

# Compute the sensitivity analysis
sens(df, group, frisked, p, "a", 0.1)

# Search over a finer grid
sens(df, group, frisked, p, "a", 0.1, eta = 0.001)

# Increase the accuracy of the maximization approximation
sens(df, group, frisked, p, "a", 0.1, m = 1001)


# Calculate 90% pointwise confidence intervals
sens(df, group, frisked, p, "a", 0.1, N = 1000, alpha = 0.1)

# Run in parallel, adjusting the chunk size to avoid blocking
sens(df, group, frisked, p, "a", 0.1, n_threads = 2, eta = 0.0001,
     chunk_size = 1000)
```

Index