Package 'prepost'

July 7, 2025

Type Package

Title Non-Parametric Bounds and Gibbs Sampler for Assessing Priming
and Post-Treatment Bias

Version 0.3.0

URL <https://github.com/mattblackwell/prepost>,

<https://mattblackwell.github.io/prepost/>

Description A set of tools to implement the non-parametric bounds and Bayesian methods for assessing post-treatment bias developed in Blackwell, Brown, Hill, Imai, and Yamamoto (2025) <doi:10.1017/pan.2025.3>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports gtools, BayesLogit, lpSolve, progress, Rglpk

Suggests testthat (>= 3.0.0), devtools, knitr, rmarkdown, lmtest,
sandwich, dplyr, ggplot2

Depends R (>= 2.10)

VignetteBuilder knitr

RoxygenNote 7.3.2

NeedsCompilation no

Author Matthew Blackwell [aut, cre],
Jacob Brown [aut],
Sophie Hill [aut],
Kosuke Imai [aut],
Teppei Yamamoto [aut]

Maintainer Matthew Blackwell <mblackwell@gmail.com>

Repository CRAN

Date/Publication 2025-07-07 12:50:02 UTC

Contents

---

delponte   *Replication dataset from Del Ponte (2020)*

Description

Ponte, Alessandro Del. "The influence of foreign elite rhetoric: National identity, emotions, and attitudes toward austerity." European Union Politics 22.1 (2021): 155-178.

Usage

    delponte

Format

A data frame with 707 rows and 53 variables:

angry_bin Self-reported anger after reading the the article

itaid_bin Binary measure of strength of Italian identity

t_commonality t=Treatment indicator for if the article praises (1) or blames (0) Italy

north Indicator for respondent living in Northern Italy

satisf Answer on a 4-point scale (rescaled to 0-1) to the question "In general, how satisfied or dissatisfied are you with your economic situation?"

sopscale Measure of how politically sophisticated the respondent is by correct answers to 2 factual question. Rescaled to 0-1.

Corriere Indicator for if the respondent is a regular reader of the Corriere blog where the survey was advertised.

Details

A dataset containing the results of an online survey experiment conducted on Italian respondents who read an article about

Source

---

land_experiment   *Replication dataset from Horowitz and Klaus (2020)*

Description

Horowitz, J., Klaus, K. "Can Politicians Exploit Ethnic Grievances? An Experimental Study of Land Appeals in Kenya." Political Behavior 42, 35–58 (2020). https://doi.org/10.1007/s11109-018-9485-1

Usage

land_experiment

Format

A data frame with 375 rows and 9 variables:

support Support for the hypothetical candidate making the appeal

land_insecure Moderator of interest measuring if the respondent is land insecure

treat_comb Treatment indicator for land-based appeals

prepost Indicator for if the moderator was measured before (0) or after(1) treatment

age Age of the respondent

female Indicator for if the respondent identifies as female

close_own Indicator for if the respondent feels close to their own ethnic group

educ Level of education (1=no schooling, 2 = some primary, 3 = complete primary, 4 = some secondary, 5 = completed secondary, 6 = college, 7 = some university, 8 = completed university),

treat Original treatment variable indicating control (0),land-based appeal (1), and land- and ethnic-based appeals (2).

Details

A dataset containing the results of an experimental study of political appeals about land greivances on candidate support in Kenya

Source

---

post_bounds   *Run post-treatment bounds.*

Description

Run post-treatment bounds.

Usage

```
post_bounds(
  formula,
  data,
  moderator,
  sims = 1000,
  conf_level = 0.95,
  moderator_mono = NULL,
  stable_mod = FALSE,
  nondiff = FALSE,
  progress = TRUE
)
```

Arguments

formula	A formula with syntax y ~ t, where y is the (unquoted) name of the outcome and t is the (unquoted) name of the treatment.
data	A data.frame containing variables in the formula, moderator, and covariates arguments.
moderator	A one-sided formuala with syntax ~ d, where d is the (unquoted) name of the moderator variable for the CATE.
sims	An integer indicating the number of simulations for the bootstrap confidence intervals for the bounds.
conf_level	A numeric indicating the confidence level for the bootstrap confidence intervals.
moderator_mono	A integer or vector of length 2 indicating if the bounds should assume monotonicity of the effect of the post-test on the moderator with 1 indicating that the post-test effect is positive and -1 indicating that it is negative. The vector of length 2 allows the monotonicity assumption to vary by treatment status with the first entry being for control and the second for treated.
stable_mod	A logical value indicating if the bounds should assume that the moderator is unaffected by pre-vs-post measurement under the control condition.
nondiff	A logical value indicating if the bounds should assume the treatment effect on the moderator is independent of the potential outcomes.
progress	A logical indicating if progress bars should be displayed. Defaults to TRUE.

Value

A list object containing bounds.

Examples

```
data(delponte)
post_bounds(
  formula = angry_bin ~ t_commonality,
   data = delponte,
  moderator = ~ itaid_bin,
 sims = 50
)
```

---

post_sens   *Run sensitivity analysis on post-measurement design*

Description

Run sensitivity analysis on post-measurement design

Usage

```
post_sens(
  formula,
  data,
  moderator,
  g_by,
  g_max = 1,
  q_by,
  sims = 1000,
  conf_level = 0.95,
  moderator_mono = NULL,
  stable_mod = FALSE,
  progress = TRUE,
  solver = "Rglpk"
)
```

Arguments

formula	A formula with syntax y ~ t, where y is the (unquoted) name of the outcome and t is the (unquoted) name of the treatment.
data	A data.frame containing variables in the formula, moderator, and covariates arguments.
moderator	A one-sided formuala with syntax ~ d, where d is the (unquoted) name of the moderator variable for the CATE.
g_by	Numeric indicating the grid spacing for the γ parameter that places an upper bound on the proportion of units whose moderator is affected by treatment.
g_max	Numeric indicating the maximum value of the γ parameter.
q_by	Numeric indicating the grid spacing for the mean of the moderator under a pre-test measurement.

sims	An integer indicating the number of simulations for the bootstrap confidence intervals for the bounds.
conf_level	A numeric indicating the confidence level for the bootstrap confidence intervals.
moderator_mono	A integer or vector of length 2 indicating if the bounds should assume monotonicity of the effect of the post-test on the moderator with 1 indicating that the post-test effect is positive and -1 indicating that it is negative. The vector of length 2 allows the monotonicity assumption to vary by treatment status with the first entry being for control and the second for treated.
stable_mod	A logical value indicating if the bounds should assume that the moderator is unaffected by pre-vs-post measurement under the control condition.
progress	A logical indicating if progress bars should be displayed. Defaults to TRUE.
solver	A character indicating what linear programming solver to use: "Rglpk" (the default) or "lpSolve".

Value

A list object containing sensitivity output.

Examples

```
data(delponte)
post_sens(formula = angry_bin ~ t_commonality,
  data = delponte,
  moderator = ~ itaid_bin,
  g_by = 0.1,
  sims = 50
)
```

---

  prepost_bounds   *Run Prepost bounds*

Description

Run Prepost bounds

Usage

```
prepost_bounds(
  formula,
  data,
  moderator,
  prepost,
  sims = 1000,
  conf_level = 0.95,
  moderator_mono = NULL,
  outcome_mono = NULL,
```

```
      stable_mod = FALSE,
      tau = 0.25,
      progress = TRUE,
      solver = "Rglpk"
    )
```

Arguments

formula	A formula with syntax y ~ t, where y is the (unquoted) name of the outcome and t is the (unquoted) name of the treatment.
data	A data.frame containing variables in the formula, moderator, and covariates arguments.
moderator	A one-sided formuala with syntax ~ d, where d is the (unquoted) name of the moderator variable for the CATE.
prepost	A one-sided formula with syntax ~ z, where z is the indicator variable for whether the moderator was measured pre- or post-treatment.
sims	An integer specifying the number of bootstrap replications for the confidence intervals.
conf_level	A numeric specifying level for the confidence intervals.
moderator_mono	A integer or vector of length 2 indicating if the bounds should assume monotonicity of the effect of the post-test on the moderator with 1 indicating that the post-test effect is positive and -1 indicating that it is negative. The vector of length 2 allows the monotonicity assumption to vary by treatment status with the first entry being for control and the second for treated.
outcome_mono	A integer or vector of length 2 indicating if the bounds should assume monotonicity of the effect of the post-test on the outcome with 1 indicating that the post-test effect is positive and -1 indicating that it is negative. The vector of length 2 allows the monotonicity assumption to vary by treatment status with the first entry being for control and the second for treated.
stable_mod	A logical value indicating if the bounds should assume that the moderator is unaffected by pre-vs-post measurement under the control condition.
tau	A numeric indicating how close the the moment conditions of the estimated bounds have to be from the minimum values in the sample. This allows us to obtain bounds and confidence intevals even when the assumptions are slightly violated due to sampling.
progress	A logical indicating if progress bars should be displayed. Defaults to TRUE.
solver	A character indicating what linear programming solver to use: "Rglpk" (the default) or "lpSolve".

Value

A list object containing bounds.

Examples

```
data(land_experiment)
prepost_bounds(
  support ~ treat_comb,
  data = land_experiment,
  moderator = ~ land_insecure,
  prepost = ~ prepost,
  sims = 50
)
```

---

prepost_gibbs   *Run Gibbs sampler for the random moderator placement design*

Description

Run Gibbs sampler for the random moderator placement design

Usage

```
prepost_gibbs(
  formula,
  data,
  prepost,
  moderator,
  covariates,
  iter = 1000,
  thin = 1,
  burn = 0,
  offset = 0,
  monotonicity = TRUE,
  stable = TRUE,
  saturated = TRUE,
  priors
)
```

Arguments

formula	A formula with syntax y ~ t, where y is the (unquoted) name of the outcome and t is the (unquoted) name of the treatment.
data	A data.frame containing variables in the formula, moderator, and covariates arguments.
prepost	A one-sided formula with syntax ~ z, where z is the indicator variable for whether the moderator was measured pre- or post-treatment.
moderator	A one-sided formuala with syntax ~ d, where d is the (unquoted) name of the moderator variable for the CATE.

covariates	A one-sided formula with syntax ~ x1 + x2, where the right-hand side variables signify which covariates the Gibbs will use to try and narrow the bounds.
iter	Integer indicating the number of iterations for the Gibbs sampler.
thin	Integer indicating how often the Gibbs sampler should save a draw.
burn	Integer indicating how many iterations should be performed before saving draws in the Gibbs sampler.
offset	A numeric value indicating the center of the prior distribution for the covariate coefficients.
monotonicity	A logical signifying whether the model assumes monotonicity.
stable	A logical signifying whether the model assumes that the the pre vs post indicator does not affect the moderator under the control condition for treatment.
saturated	A logical indicating whether the coefficients on the covariates are allowed to vary by the principal strata.
priors	A list object containing the priors for the Gibbs sampler. Priors include beta.precision, psi.precision, alpha, y.alpha, and y.beta.

Value

A list object containing Gibbs posterior quantities of interest and parameters.

Examples

```
data(land_experiment)
prepost_gibbs(
  support ~ treat_comb,
  data = land_experiment,
  moderator = ~ land_insecure,
  prepost = ~ prepost,
  covariates = ~ educ + age,
  iter = 50
)
```

---

prepost_gibbs_nocovar   *Run Gibbs sampler without covariates*

Description

Run Gibbs sampler without covariates

Usage

```
prepost_gibbs_nocovar(
  formula,
  data,
  prepost,
  moderator,
```

```
  iter = 1000,
  thin = 1,
  burn = 0,
  monotonicity = TRUE,
  stable = TRUE,
  priors,
  predictive = FALSE
)
```

Arguments

`formula`	A formula with syntax y ~ t, where y is the name of the outcome variable and t is the name of the treatment.
`data`	A data frame containin the variables in the formula.
`prepost`	A one-sided formula with syntax ~ z, where z is the indicator variable for whether the moderator was measured pre- or post-treatment.
`moderator`	A formuala with syntax ~ d, where d is the moderator variable for the CATE.
`iter`	Numeric, number of iterations for the Gibbs
`thin`	Numeric, thinning parameter for the Gibbs
`burn`	Numeric, burn in rate for the Gibbs
`monotonicity`	A logical signifying whether Gibbs assumes monotonicity.
`stable`	A logical signifying whether Gibbs assumes stability.
`priors`	A list object containing the priors for the Gibbs sampler. Priors include beta.precision, psi.precision, alpha, y.alpha, and y.beta.
`predictive`	A logical indicator for whether to return prior predictive draws (TRUE) or posterior draws (FALSE, default).

Value

A list object containing Gibbs posterior quantities of interest and parameters.

Examples

```
data(land_experiment)
prepost_gibbs_nocovar(
  support ~ treat_comb,
  data = land_experiment,
  moderator = ~ land_insecure,
  prepost = ~ prepost,
  iter = 50
)
```

---

prepost_sens	*Run sensitivity analysis for the randomized moderator placement design*

Description

Run sensitivity analysis for the randomized moderator placement design

Usage

```
prepost_sens(
  formula,
  data,
  moderator,
  prepost,
  g_by,
  g_at,
  t_by,
  t_at,
  sims = 1000,
  stable_mod = FALSE,
  conf_level = 0.95,
  moderator_mono = NULL,
  outcome_mono = NULL,
  tau = 0.25,
  progress = TRUE,
  solver = "Rglpk"
)
```

Arguments

formula	A formula with syntax y ~ t, where y is the (unquoted) name of the outcome and t is the (unquoted) name of the treatment.
data	A data.frame containing variables in the formula, moderator, and covariates arguments.
moderator	A one-sided formuala with syntax ~ d, where d is the (unquoted) name of the moderator variable for the CATE.
prepost	A one-sided formula with syntax ~ z, where z is the indicator variable for whether the moderator was measured pre- or post-treatment.
g_by	Numeric indicating the grid spacing for the γ parameter that places an upper bound on the proportion of units whose moderator is affected by treatment.
g_at	Vector specifying what values to set the γ parameter to in the sensitivity analysis. Overrides g_by.
t_by	Numeric indicating the grid spacing for the θ parameter that restricts what proportion of units have their outcomes affected by the pre vs post-measurement of the moderator.

t_at   Vector specifying what values to set the θ parameter to in the sensitivity analysis.
                 Overrides `t_by`.

sims   An integer indicating the number of simulations for the bootstrap confidence
                 intervals for the bounds.

stable_mod   A logical value indicating if the bounds should assume that the moderator is
                 unaffected by pre-vs-post measurement under the control condition.

conf_level   A numeric indicating the confidence level for the bootstrap confidence intervals.

moderator_mono   A integer or vector of length 2 indicating if the bounds should assume mono-
                 tonicity of the effect of the post-test on the moderator with 1 indicating that the
                 post-test effect is positive and `-1` indicating that it is negative. The vector of
                 length 2 allows the monotonicity assumption to vary by treatment status with
                 the first entry being for control and the second for treated.

outcome_mono   A integer or vector of length 2 indicating if the bounds should assume mono-
                 tonicity of the effect of the post-test on the outcome with 1 indicating that the
                 post-test effect is positive and `-1` indicating that it is negative. The vector of
                 length 2 allows the monotonicity assumption to vary by treatment status with
                 the first entry being for control and the second for treated.

tau   A numeric indicating how close the the moment conditions of the estimated
                 bounds have to be from the minimum values in the sample. This allows us to
                 obtain bounds and confidence intevals even when the assumptions are slightly
                 violated due to sampling.

progress   A logical indicating if progress bars should be displayed. Defaults to TRUE.

solver   A character indicating what linear programming solver to use: "Rglpk" (the
                 default) or "lpSolve".

Value

A list object containing sensitivity output.

Examples

```
data(land_experiment)
prepost_sens(
  support ~ treat_comb,
  data = land_experiment,
  moderator = ~ land_insecure,
  prepost =  ~ prepost,
  g_by = 0.1,
  t_at = c(0.25, 1),
  sims = 50,
  moderator_mono = NULL
)
```

---

pre_bounds   *Run pre-treatment bounds.*

Description

Run pre-treatment bounds.

Usage

```
pre_bounds(formula, data, moderator, conf_level = 0.95, outcome_mono = 1L)
```

Arguments

formula	A formula with syntax y ~ t, where y is the (unquoted) name of the outcome and t is the (unquoted) name of the treatment.
data	A data.frame containing variables in the formula, moderator, and covariates arguments.
moderator	A one-sided formuala with syntax ~ d, where d is the (unquoted) name of the moderator variable for the CATE.
conf_level	A numeric indicating the confidence level for the bootstrap confidence intervals.
outcome_mono	A integer indicating the direction of the priming monotonicity assumption. The default value 1 indicates that asking the moderator question in the pre-test moves outcomes in a positive direction for all units. The value -1 indicates it moves outcomes in a negative direction for all units.

Value

A list object containing bounds.

Examples

```
data(delponte)
pre_bounds(
  formula = angry_bin ~ t_commonality,
   data = delponte,
  moderator = ~ itaid_bin
)
```

---

pre_sens   *Run sensitivity analysis on pre-test design*

Description

Run sensitivity analysis on pre-test design

Usage

```
pre_sens(
  formula,
  data,
  moderator,
  t_by = 0.05,
  conf_level = 0.95,
  outcome_mono = 1L
)
```

Arguments

formula	A formula with syntax y ~ t, where y is the (unquoted) name of the outcome and t is the (unquoted) name of the treatment.
data	A data.frame containing variables in the formula, moderator, and covariates arguments.
moderator	A one-sided formuala with syntax ~ d, where d is the (unquoted) name of the moderator variable for the CATE.
t_by	Numeric indicating the grid spacing for the θ parameter that restricts what proportion of units have their outcomes affected by the pre vs post-measurement of the moderator.
conf_level	A numeric indicating the confidence level for the bootstrap confidence intervals.
outcome_mono	A integer indicating the direction of the priming monotonicity assumption. The default value 1 indicates that asking the moderator question in the pre-test moves outcomes in a positive direction for all units. The value -1 indicates it moves outcomes in a negative direction for all units.

Value

A list object containing sensitivity output.

Examples

```
pre_sens(formula = angry_bin ~ t_commonality,
  data = delponte,
  moderator = ~ itaid_bin,
  t_by = 0.1
)
```

Index