

Package: ivrobustness (via r-universe)

October 16, 2026

Title Robustness Values for Causal Estimation

Version 0.0.0.9000

Description What the package does (one paragraph).

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/rextendr/version 0.5.0

SystemRequirements Cargo (Rust's package manager), rustc >= 1.89.0, xz

Depends R (>= 4.2)

Suggests RTMB, testthat (>= 3.0.0)

Config/testthat/edition 3

URL <http://archive.beniamino.org/ivrobustness/>

Imports dplyr, purrr, rlang, stringr, tibble

Config/pak/sysreqs libicu-dev xz-utils libclang-dev

Repository <https://beniaminogreen.r-universe.dev>

Date/Publication 2026-09-16 19:25:43 UTC

RemoteUrl <https://github.com/beniaminogreen/ivrobustness>

RemoteRef HEAD

RemoteSha 8e6e1b664814d87ca4ac28afb9ba45319644b19

Contents

benchmark_covariates	2
interpret_table	2
robustness_values	3

Index	5
--------------	----------

benchmark_covariates	<i>Benchmark Robustness Values Based on Observed Covariates</i>
----------------------	---

Description

Benchmark Robustness Values Based on Observed Covariates

Usage

```
benchmark_covariates(w, z, y, X)
```

Arguments

w	the instrument vector, a numeric (double) vector with no NA values
z	the treatment vector, a numeric (double) vector with no NA values
y	the outcome vecctor, a numeric (double) vector with no NA values
X	a matrix of control covaraites used for benchmarking

Value

a dataframe containing the robustness values associated with held out covariates

interpret_table	<i>Interpret a Table Of Robustness Values</i>
-----------------	---

Description

Struggling to interpret the results of robustness_values? Look no further! interpret_table helps by providing a plain-text interpretation of the robustness values of a design. It interprets each of the four robustness values for you, and flags low robustness values.

Usage

```
interpret_table(df)
```

Arguments

df	a datasdrame, as returned by robustness_values
----	--

Value

A string

Examples

```

create_simulation <- function(n=10^7, rho = .5) {
  w <- rnorm(n)
  z <- sqrt(rho)*w + sqrt(1-rho)*rnorm(n)
  y <- 2*z + rnorm(n)

  return(list(
    w = w,
    z = z,
    y = y
  ))
}

weak_instrument_data <- create_simulation(rho = .01)

df <- robustness_values(
  weak_instrument_data$w,
  weak_instrument_data$z,
  weak_instrument_data$y
)
cat(interpret_table(df))

```

robustness_values	<i>Calculate Robustness Values</i>
-------------------	------------------------------------

Description

Calculate Robustness Values

Usage

```
robustness_values(w, z, y, X = NULL)
```

Arguments

w	the instrument vector, a numeric (double) vector with no NA values
z	the treatment vector, a numeric (double) vector with no NA values
y	the outcome vecctor, a numeric (double) vector with no NA values
X	an optional matrix of covariates to residualize on

Value

a dataframe containing robustness values with the following columns:

- *Method* The method for the associated Robustness Value:
 - SOO - Selection on observables
 - IV - Instrumental variables

- ExogenousIV - Instrumental variables, where exogeneity is known to be satisfied (as in a randomized experiment)
- ExcludableIV - Instrumental variables, where the exclusion restriction is known to be satisfied
- *RV* the robustness value
- *rho_1, rho_2, rho_3* The parameters used to achieve this robustness value
- *bias* The bias achieved at this combination of parameters
- *assumption gap* the largest gap between partial correlations that are assumed to be zero (only relevant for ExogenousIV and ExcludableIV) and the actual value found by the optimizer. This should be negligible and close to the machine tolerance.

Examples

```
create_simulation <- function(n=10^7, rho = .5) {
  w <- rnorm(n)
  z <- sqrt(rho)*w + sqrt(1-rho)*rnorm(n)
  y <- 2*z + rnorm(n)

  return(list(
    w = w,
    z = z,
    y = y
  ))
}

strong_instrument_data <- create_simulation(rho = .90)
weak_instrument_data <- create_simulation(rho = .01)

robustness_values(
  strong_instrument_data$w,
  strong_instrument_data$z,
  strong_instrument_data$y
)

robustness_values(
  weak_instrument_data$w,
  weak_instrument_data$z,
  weak_instrument_data$y
)
```

Index

benchmark_covariates, [2](#)

interpret_table, [2](#)

robustness_values, [3](#)