Package: formulaiv (via r-universe)

October 24, 2026

Title Sensitivity of Formula Instrument to Shock Design

Version 0.1.0

Description Functions to implement the formula instrument method in Borusyak and Hull (2023) <doi:10.3982/ECTA19367> and examine its sensitivity to the assumed distributional of counterfactual shocks.

License MIT + file LICENSE

URL https://github.com/peizansheng/formulaiv

BugReports https://github.com/peizansheng/formulaiv/issues

Encoding UTF-8

LazyData true

LazyDataCompression xz

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports stats, lpSolve

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

Depends R (>= 3.5)

VignetteBuilder knitr

Repository https://peizansheng.r-universe.dev

Date/Publication 2026-06-18 18:44:01 UTC

RemoteUrl https://github.com/peizansheng/formulaiv

RemoteRef HEAD

RemoteSha 7d96a946b14a30d0053499672afba1e491bc9c66

Contents

BH_precomputed_results 2
formulaiv 2
line 4
ma 5

Index **6**

`BH_precomputed_results` *Precomputed BH (2023) Sensitivity Bounds*

Description

A named list of four `formulaiv()` bound tables for the Borusyak and Hull (2023) China market-access application, used as the expected values in `tests/testthat/test-bh2023.R`. Each element is the `$beta` data frame returned by `formulaiv()` for one specification. Because they are produced by the package itself, that test is a self-regression check; regenerate the list with `source("data-raw/BH_precomputed_results.R")`.

Usage

```
BH_precomputed_results
```

Format

A named list of four data frames:

`BH_sens_joint_cons_no_controls` joint set, no controls, `eps = seq(1, 20, 0.2)`

`BH_sens_joint_cons_with_controls` joint set, with controls, `eps = seq(1, 20, 0.2)`

`BH_sens_marginal_cons_no_controls` marginal set, no controls, `eps = seq(1, 2.5, 0.25)`

`BH_sens_marginal_cons_with_controls` marginal set, with controls, `eps = seq(1, 2.5, 0.25)`

Each data frame has columns `eps`, `lb`, and `ub` (the sensitivity bounds).

`formulaiv` *Sensitivity of formula instrument to shock design*

Description

`formulaiv()` evaluates the sensitivity of formula instrument estimates to small or large deviations away from an assumed baseline distribution of shocks.

Usage

```
formulaiv(y, x, z, f, eps, cons, controls = NULL, denom_tol = 1e-08)
```

Arguments

`y`	outcome ($N \times 1$ vector)
`x`	endogenous regressor / treatment ($N \times 1$ vector)
`z`	formula instrument ($N \times 1$ vector)
`f`	counterfactual shock realizations, one scenario per column ($N \times S$ data frame or matrix)
`eps`	degree of sensitivity of interest ($M \times 1$ vector; a scalar is allowed), where `eps` corresponds to • κ for joint constraints • δ for marginal constraints
`cons`	constraint list specifying the sensitivity set. One of: • **joint** — `list(name = "joint", pbar = pbar)`, where `pbar` is the baseline distribution over the `S` scenarios ($S \times 1$ vector). • **marginal** — `list(name = "marginal", g = g, qbar = qbar)` *or* `list(name = "marginal", g = g, pbar = pbar)`, where – `g` is the shock realization matrix ($L \times S$ data frame or matrix) whose row marginals are constrained, and – the baseline marginals are supplied either directly as `qbar` ($L \times 1$ vector) or derived from a baseline joint distribution `pbar` ($S \times 1$ vector) via $\bar{q} = g\,\bar{p}$.
`controls`	control variables ($N \times J$ data frame or matrix, or `NULL`)
`denom_tol`	numeric tolerance below which the first-stage denominator is treated as zero when reading off its sign over the feasible set (default `1e-8`). This is an internal numerical safeguard that rarely needs changing.

Value

A list of two data frames:

- `beta` — the sensitivity bounds, one row per `eps` (columns `eps`, `lb`, `ub`).
- `p_opt` — the optimal weights achieving each bound, in tidy long form (columns `eps`, `scenario`, `lb`, `ub`; `MS` rows).

Numerical conditioning

Each bound is the optimal value of a linear program solved with `lpSolve::lp()`, whose simplex routine is sensitive to the magnitude of the problem coefficients. Inputs on a very large scale — for example population- or weight-aggregated sums — can make the solver report a numerical failure (`lpSolve` status `5`) instead of returning a bound. The estimate is invariant to multiplying `y` and `x` by a common positive constant, so when this happens it is enough to divide both `y` and `x` by a single scale factor that brings them to within a few orders of magnitude of `1`: every bound (`beta$lb`, `beta$ub`) is unchanged. (Rescaling only one of `y` or `x`, or rescaling them by different factors, *does* change the estimate.)

Examples

```
# BH market-access data (bundled with the package)
y <- ma$emp_growth # outcome (N x 1)
x <- ma$dma0 # endogenous regressor (N x 1)
z <- x # formula instrument (N x 1)
f <- ma[, paste0("ma_nlink", 1:1999)] - ma$ma2007 # shock draws (N x S)
pbar <- rep(1 / 1999, 1999) # baseline weights (S x 1)

# Joint sensitivity set without controls, evaluated at eps = 1.5 and 2.
# Wrapped in \donttest{} because solving the linear-fractional programs over
# all S = 1999 shock draws takes longer than CRAN's 5s example budget.

formulaiv(
  y = y,
  x = x,
  z = z,
  f = f,
  eps = c(1.5, 2),
  cons = list(name = "joint", pbar = pbar)
)$beta
```

`line` *High Speed Railway (HSR) Data*

Description

high speed railway (HSR) data from Borusyak and Hull (2023) replication file

Usage

```
line
```

Format

A data frame with 150 rows and 2022 columns. The package directly uses the following groups of columns:

- line characteristics used in the ML model: `line_type_en`, `speed`, `computed_length`, `nlinks`, `year_approved`, `plan_type`
- realized opening year: `year_operate0`
- simulated opening years: `year_operate1` to `year_operate1999`

Source

https://zenodo.org/records/8286785

ma *Market Access (MA) Data*

Description

city-level data from Borusyak and Hull (2023) replication file

Usage

```
ma
```

Format

A data frame with 275 rows and 2013 columns. The sensitivity analysis uses the following groups of columns:

- identifiers and geography: `cityid`, `year`, `latitude`, `longitude`, `scaled_lat`, `scaled_lon`, `distance_B`
- outcomes and baseline market access: `emp_growth`, `dma0`, `ma0`, `ma2007`
- market-access simulations: `ma_nlink1` to `ma_nlink1999`
- expected market access: `ma_nlink_pscore`, `dma_nlink_pscore`
- recentered market access: `ma_nlink_rc`

Source

https://zenodo.org/records/8286785

Index

∗ **datasets**
BH_precomputed_results, 2
line, 4
ma, 5

BH_precomputed_results, 2

formulaiv, 2
formulaiv(), *2*

line, 4
lpSolve::lp(), *3*

ma, 5