

Package: ssBartik (via r-universe)

July 19, 2026

Type Package

Title End-to-End Pipeline for Shift-Share (Bartik) Instrumental Variables

Version 0.1.1

Description Construction, diagnostics, estimation, exposure-robust inference and publication-ready visualisation for shift-share (Bartik) instrumental variable designs, in one consistent workflow. Organised around the two identification routes of the modern literature. The exogenous share approach refers to Goldsmith-Pinkham, Sorkin and Swift (2020) <[doi:10.1257/aer.20181047](https://doi.org/10.1257/aer.20181047)> and the exogenous shift approach refers to Borusyak, Hull and Jaravel (2022) <[doi:10.1093/restud/rdab030](https://doi.org/10.1093/restud/rdab030)> and Adao, Kolesar and Morales (2019) <[doi:10.1093/qje/qjz025](https://doi.org/10.1093/qje/qjz025)>. Wraps 'ShiftShareSE' for exposure-robust inference when available.

License MIT + file LICENSE

Encoding UTF-8

Imports ggplot2 (>= 3.5.0), grDevices, grid, stats, utils

Suggests ShiftShareSE, testthat (>= 3.0.0)

URL <https://github.com/takuma1102/ssBartik>,
<https://takuma1102.github.io/ssBartik/>

BugReports <https://github.com/takuma1102/ssBartik/issues>

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Takuma Iwasaki [aut, cre] (affiliation: Stanford University)

Maintainer Takuma Iwasaki <iwasakit@stanford.edu>

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

Date/Publication 2026-07-19 12:30:02 UTC

RemoteUrl <https://github.com/cran/ssBartik>

RemoteRef HEAD

RemoteSha d5133a978d8bfd748b9ee116cc87c69d51573db9

Contents

format.ssb_drop_top	3
format.ssb_estimate	3
format.ssb_loo	4
format.ssb_overid	5
format.ssb_rottemberg	6
format.ssb_shock_balance	7
format.ssb_shocks	7
format.ssb_weight_summary	8
plot.ssb_rottemberg	9
ssb_aggregate	10
ssb_design	11
ssb_drop_top	12
ssb_equivalence	13
ssb_estimate	14
ssb_first_stage	15
ssb_loo	16
ssb_overid	17
ssb_pipeline	18
ssb_placebo	19
ssb_plot_ci	20
ssb_plot_loo	20
ssb_plot_overid	21
ssb_plot_ri	22
ssb_plot_rottemberg	22
ssb_plot_shocks	23
ssb_pretrend	23
ssb_recenter	24
ssb_ri	25
ssb_rottemberg	26
ssb_share_balance	27
ssb_shock_balance	28
ssb_shock_iv	28
ssb_shock_summary	29
ssb_table_image	29
ssb_weight_summary	30
ssbartik	31

Index

34

format.ssb_drop_top *Render a drop-top-shocks comparison as LaTeX or Markdown*

Description

Paste-ready full-vs-reduced table from [ssb_drop_top()].

Usage

```
## S3 method for class 'ssb_drop_top'
format(
  x,
  output = c("latex", "markdown"),
  digits = 3,
  caption = NULL,
  label = "tab:ssb-droptop",
  ...
)
```

Arguments

x	An [ssb_drop_top()] object.
output	"latex" (booktabs) or "markdown" (pipe table).
digits	Decimal places for the estimate, SE and interval.
caption, label	Table caption and cross-reference label (LaTeX only).
...	Unused.

Value

A character vector of the table lines; pass to 'writeLines()'.

format.ssb_estimate *Render the estimate / standard-error table as LaTeX or Markdown*

Description

Turns an [ssb_estimate()] table into a publication-ready comparison of the point estimate across standard-error methods. "latex" uses **booktabs** rules; "markdown" emits a GitHub pipe table. Rows whose standard error is unavailable are dropped. Mirrors [format.ssb_rottemberg()].

Usage

```
## S3 method for class 'ssb_estimate'
format(
  x,
  output = c("latex", "markdown"),
  digits = 3,
  caption = NULL,
  label = "tab:ssb-estimate",
  ...
)
```

Arguments

<code>x</code>	An [ssb_estimate()] object.
<code>output</code>	"latex" (booktabs) or "markdown" (pipe table).
<code>digits</code>	Decimal places for the estimate, SE and interval.
<code>caption, label</code>	Table caption and cross-reference label (LaTeX only).
<code>...</code>	Unused.

Value

A character vector of the table lines (paste-ready); pass to `writeLines()`.

format.ssb_loo	<i>Render a leave-one-out table as LaTeX or Markdown</i>
----------------	--

Description

Paste-ready version of the [ssb_loo()] sensitivity table (one row per dropped shock, with the re-estimated coefficient); the overall estimate is in a note.

Usage

```
## S3 method for class 'ssb_loo'
format(
  x,
  output = c("latex", "markdown"),
  digits = 3,
  caption = NULL,
  label = "tab:ssb-loo",
  ...
)
```

Arguments

x	An [ssb_loo()] object.
output	"latex" (booktabs) or "markdown" (pipe table).
digits	Decimal places for the estimate, SE and interval.
caption, label	Table caption and cross-reference label (LaTeX only).
...	Unused.

Value

A character vector of the table lines; pass to 'writeLines()'.

format.ssb_overid	<i>Render an overidentification test as LaTeX or Markdown</i>
-------------------	---

Description

Paste-ready statistic/value table for the [ssb_overid()] cross-instrument homogeneity test.

Usage

```
## S3 method for class 'ssb_overid'
format(
  x,
  output = c("latex", "markdown"),
  digits = 3,
  caption = NULL,
  label = "tab:ssb-overid",
  ...
)
```

Arguments

x	An [ssb_overid()] object.
output	"latex" (booktabs) or "markdown" (pipe table).
digits	Decimal places for the estimate, SE and interval.
caption, label	Table caption and cross-reference label (LaTeX only).
...	Unused.

Value

A character vector of the table lines; pass to 'writeLines()'.

format.ssb_rotemberg	<i>Render the Rotemberg-weight table as paste-ready LaTeX or Markdown</i>
----------------------	---

Description

Turns the `[ssb_rotemberg()]` decomposition into a publication-quality table of the top-weight shocks. The `"latex"` output uses **booktabs** rules and math-mode headers (load the booktabs LaTeX package); `"markdown"` emits a GitHub pipe table. Both list, per shock, the Rotemberg weight $\hat{\alpha}_n$, its just-identified estimate $\hat{\beta}_n$, the first-stage F and the shock g_n , with the overall estimate and the positive/negative weight sums in a note.

Usage

```
## S3 method for class 'ssb_rotemberg'
format(
  x,
  output = c("latex", "markdown"),
  n = 6,
  digits = 3,
  caption = NULL,
  label = "tab:rotemberg",
  ...
)
```

Arguments

<code>x</code>	An <code>[ssb_rotemberg()]</code> object.
<code>output</code>	<code>"latex"</code> (booktabs) or <code>"markdown"</code> (pipe table).
<code>n</code>	Number of top-weight shocks to include.
<code>digits</code>	Decimal places for the estimates and weights.
<code>caption, label</code>	Table caption and cross-reference label (LaTeX only).
<code>...</code>	Unused.

Value

A character vector of the table lines (paste-ready); pass to `'writeLines()'`.

See Also

`[plot.ssb_rotemberg()]` for a rendered image, `[ssb_plot_rotemberg()]` for the bubble figure.

format.ssb_shock_balance	<i>Render a shock-balance test as LaTeX or Markdown</i>
--------------------------	---

Description

Paste-ready coefficient table for the [ssb_shock_balance()] test, with the joint Wald statistic in a note.

Usage

```
## S3 method for class 'ssb_shock_balance'
format(
  x,
  output = c("latex", "markdown"),
  digits = 3,
  caption = NULL,
  label = "tab:ssb-shock-balance",
  ...
)
```

Arguments

- x An [ssb_shock_balance()] object.
- output "'latex'" (booktabs) or "'markdown'" (pipe table).
- digits Decimal places for the estimate, SE and interval.
- caption, label Table caption and cross-reference label (LaTeX only).
- ... Unused.

Value

A character vector of the table lines; pass to 'writeLines()'.

format.ssb_shocks	<i>Render a shock-exposure summary as LaTeX or Markdown</i>
-------------------	---

Description

Paste-ready version of the [ssb_shock_summary()] diagnostic: the top shocks by exposure weight, with the effective number of shocks and concentration in a note.

Usage

```
## S3 method for class 'ssb_shocks'
format(
  x,
  output = c("latex", "markdown"),
  digits = 3,
  caption = NULL,
  label = "tab:ssb-shocks",
  top = 8,
  ...
)
```

Arguments

x	An [ssb_shock_summary()] ('ssb_shocks') object.
output	"latex" (booktabs) or "markdown" (pipe table).
digits	Decimal places for the estimate, SE and interval.
caption, label	Table caption and cross-reference label (LaTeX only).
top	Number of top-exposure shocks to include.
...	Unused.

Value

A character vector of the table lines; pass to 'writeLines()'.

```
format.ssb_weight_summary
```

Render a Rotemberg-weight summary as LaTeX or Markdown

Description

Paste-ready version of the [ssb_weight_summary()] table (top shocks by weight), with the largest weight and the weight/estimate/F correlations in a note.

Usage

```
## S3 method for class 'ssb_weight_summary'
format(
  x,
  output = c("latex", "markdown"),
  digits = 3,
  caption = NULL,
  label = "tab:ssb-weights",
  ...
)
```

Arguments

x	An [ssb_weight_summary()] object.
output	"latex" (booktabs) or "markdown" (pipe table).
digits	Decimal places for the estimate, SE and interval.
caption, label	Table caption and cross-reference label (LaTeX only).
...	Unused.

Value

A character vector of the table lines; pass to 'writeLines()'.

plot.ssb_rottemberg	<i>Render the Rotemberg-weight table as a compact booktabs figure</i>
---------------------	---

Description

Draws the top-weight-shock table (see [format.ssb_rottemberg()] for the columns) as a paper-style image with normal single-line row spacing. Writes to the active device, or to 'file' when supplied ('.png' default, '.pdf' for vector output). For LaTeX/Markdown source use [format()] instead; for the bubble scatter use [ssb_plot_rottemberg()].

Usage

```
## S3 method for class 'ssb_rottemberg'
plot(
  x,
  file = NULL,
  width = NULL,
  height = NULL,
  res = 200,
  n = 6,
  digits = 3,
  note = .ssb_rot_note(),
  ...
)
```

Arguments

x	An [ssb_rottemberg()] object.
file	Optional output path; the format is taken from the extension ('.png' or '.pdf').
width, height	Figure size in inches; defaults adapt to the content.
res	Resolution in PPI for the '.png' device (ignored for '.pdf').
n	Number of top-weight shocks to include.
digits	Decimal places for the estimates and weights.

note	Footnote shown left-aligned below the table (an italic "Note:" label is prepended). Defaults to a definition of the columns rendered in maths so the symbols match the headers. Pass 'NULL' to omit it, a character string for a plain note (explicit line breaks honoured, never auto-wrapped), or your own plotmath expression.
...	Unused.

Value

The object, invisibly (called for its side effect).

ssb_aggregate	<i>Aggregate a shift-share design to the shock (shifter) level</i>
---------------	--

Description

Collapses the unit-level design to one row per shock, following the Borusyak-Hull-Jaravel (2022) equivalence. With controls partialled out of the outcome and treatment (weighted FWL), each shock n gets an exposure weight $s_n = \sum_i e_i s_{in}$ and exposure-weighted means $\bar{y}_n = \sum_i e_i s_{in} \tilde{y}_i / s_n$ and $\bar{x}_n$ similarly. Running an IV of $\bar{y}_n$ on $\bar{x}_n$ with instrument g_n and weights s_n reproduces the location-level shift-share estimate exactly (see [ssb_equivalence()]).

Usage

```
ssb_aggregate(design)
```

Arguments

design An [ssb_design()] object.

Value

A 'data.frame' (class 'ssb_aggregate') with columns 'sector', 'g', 's_bar' (exposure weight), 'x_bar', 'y_bar'.

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "shift")
ssb_aggregate(d)
```

ssb_design

*Define a shift-share (Bartik) IV design***Description**

'ssb_design()' is the single entry point of the package. It takes the three pieces of a shift-share design — a unit-level table, a long table of exposure shares, and a table of shocks (shifts) — aligns them, and constructs the Bartik instrument $z_i = \sum_n s_{in} g_n$. The resulting object flows directly into diagnostics ([ssb_rottemberg()], [ssb_shock_summary()], ...), estimation ([ssb_estimate()]) and plotting.

Usage

```
ssb_design(
  data,
  shares,
  shocks,
  y = "y",
  x = "x",
  location = "location",
  sector = "sector",
  time = NULL,
  controls = NULL,
  weights = NULL,
  cluster = NULL,
  share_col = "share",
  shock_col = "shock",
  exogenous = c("shift", "share")
)
```

Arguments

data	A unit-level 'data.frame': one row per location (or location-period). Must contain 'y', 'x', 'location', and any 'controls', 'weights', 'cluster' columns referenced below.
shares	A long 'data.frame' of exposure shares with columns 'location', 'sector', the share column ('share_col'), and 'time' for panels.
shocks	A 'data.frame' of shocks with columns 'sector', the shock column ('shock_col'), and 'time' for panels.
y, x	Column names (strings) of the outcome and endogenous treatment.
location, sector	Column names of the unit and sector identifiers.
time	Optional column name of a period identifier (present in 'data', 'shares' and 'shocks') for panel designs.

controls	Optional character vector of control columns in ‘data’. Numeric columns enter linearly; factor or character columns are expanded into dummies, so period or region fixed effects can be supplied as factors (in panel shift-share designs, period fixed effects are usually essential — shocks should be compared within periods).
weights	Optional column name of regression weights in ‘data’.
cluster	Optional column name of a clustering variable in ‘data’.
share_col	Name of the exposure-share column in ‘shares’ (default “share”).
shock_col	Name of the shock (shift) column in ‘shocks’ (default “shock”).
exogenous	Which identification route to emphasise downstream: “shift” (shocks) or “share” (shares). “shock”/“shares” are accepted aliases.

Details

The **instrument** is constructed identically whichever identification route you take; the ‘exogenous’ argument only governs which **diagnostics** and **controls** are appropriate downstream (see [ssb_pipeline()]). Set ‘exogenous = “share”’ for the exogenous-shares route (Goldsmith-Pinkham, Sorkin and Swift 2020; Rotemberg-weight diagnostics) or ‘exogenous = “shift”’ for the exogenous-shocks route (Borusyak, Hull and Jaravel 2022; Adao, Kolesar and Morales 2019; shock-level diagnostics and AKM inference).

Value

An object of class ‘ssb_design’.

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "share")
d
```

ssb_drop_top

Re-estimate after dropping the top-weight shocks

Description

Removes the ‘n’ shocks with the largest absolute Rotemberg weight **together** and re-estimates, to see whether the headline result survives without the most influential shocks. (Contrast [ssb_loo()], which drops one at a time.)

Usage

```
ssb_drop_top(design, n = 5, methods = c("iid", "ehw", "akm", "akm0"))
```

Arguments

design	An [ssb_design()] object.
n	Number of top-weight shocks to drop.
methods	Inference methods for the comparison, passed to [ssb_estimate()] (defaults to the exposure-robust panel; add "cluster" / "twoway" if wanted).

Value

A list (class 'ssb_drop_top') with the 'dropped' sectors and the 'full' and 'reduced' [ssb_estimate()] tables.

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "share")
ssb_drop_top(d, n = 3)
```

ssb_equivalence	<i>Check the location-level / shock-level equivalence</i>
-----------------	---

Description

Verifies numerically that the location-level shift-share IV estimate equals the shock-level IV estimate (Borusyak-Hull-Jaravel 2022). A near-zero difference is a strong internal-consistency check that the instrument and aggregation are behaving as intended.

Usage

```
ssb_equivalence(design)
```

Arguments

design	An [ssb_design()] object.
--------	---------------------------

Value

A list (class 'ssb_equivalence') with 'location', 'shock', and their absolute 'difference'.

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "shift")
ssb_equivalence(d)
```

ssb_estimate

Estimate a shift-share IV regression with several confidence intervals

Description

Computes the shift-share 2SLS point estimate of ‘x’ on ‘y’ (instrumented by the constructed Bartik instrument, controls partialled out via FWL) and reports a panel of intervals side by side so the practical importance of the inference method is visible:

- ‘iid’ — classical (homoskedastic) IV,
- ‘ehw’ — Eicker-White (heteroskedasticity-robust),
- ‘akm’, ‘akm0’ — Aday-Kolesar-Morales exposure-robust inference, via **ShiftShareSE** when installed,
- ‘cluster’ — naive cluster-robust (needs ‘cluster’ in the design),
- ‘twoway’ — two-way cluster-robust (needs ‘cluster’ in the design and ‘cluster2’ here).

The point estimate is identical across rows; only the standard errors and intervals differ, which is exactly what makes the comparison instructive.

Usage

```
ssb_estimate(
  design,
  methods = c("ehw", "cluster", "akm", "akm0"),
  level = 0.95,
  cluster2 = NULL,
  shock_cluster = NULL
)
```

Arguments

design	An [ssb_design()] object.
methods	Which methods to report. Defaults to ‘ehw’, ‘cluster’, ‘akm’, ‘akm0’; add “iid” (homoskedastic) and/or “twoway” for two-way clustering. ‘cluster’ needs the design’s ‘cluster’ column (else ‘NA’).
level	Confidence level for the reported intervals.
cluster2	Optional second clustering column in ‘data’ for the “twoway” method (paired with the design’s ‘cluster’).
shock_cluster	Optional grouping of the shocks for the AKM / AKM0 variance: a column name in the shocks table, or a vector of length equal to the number of shock-cells. Use it when shocks are mutually correlated within groups — e.g. sub-industries within broader industries, or sector cells of the same sector across periods — so the exposure-robust variance is clustered at the group level (Aday, Kolesar & Morales 2019; passed to ShiftShareSE as ‘sector_cvar’).

Details

The primary object of the comparison is the **confidence interval**, not the standard error: AKM0 in particular is defined directly as a (possibly asymmetric, possibly unbounded) interval, and the ‘std.error’ reported for it is a symmetric pseudo-SE implied by that interval rather than a conventional standard error. Read the table and [ssb_plot_ci()] figure as a comparison of intervals. When the instrument is weak the AKM0 confidence **set** need not be an interval at all: it can be the whole real line or the complement of an interval (a union of two rays). **ShiftShareSE** encodes the latter as ‘conf.low > conf.high’; ‘ssb_estimate()’ flags both cases in the ‘note’ column and the table/plot methods render them accordingly.

The default panel is ‘ehw’ / ‘cluster’ / ‘akm’ / ‘akm0’: two classical robust SEs (heteroskedasticity- and cluster-robust) shown next to the two exposure-robust AKM / AKM0 intervals. ‘iid’ (homoskedastic) and ‘twoway’ are not in the default; request them explicitly via ‘methods’ when wanted (e.g. ‘methods = c("iid", "ehw", "cluster", "twoway", "akm", "akm0)’, adding ‘cluster2’ for two-way clustering). The ‘cluster’ row needs a ‘cluster’ column in the design (set via [ssb_design()]); without one it is reported as ‘NA’ with a note rather than an error.

Value

A ‘data.frame’ of class ‘ssb_estimate’ with one row per method (‘estimate’, ‘std.error’, ‘conf.low’, ‘conf.high’), carrying the first-stage F as an attribute. Plot with [ssb_plot_ci()].

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "share")
ssb_estimate(d)
```

ssb_first_stage	<i>First-stage strength: standard and exposure-robust (effective) F</i>
-----------------	---

Description

Reports the standard heteroskedasticity-robust first-stage F of the treatment on the constructed instrument, and an exposure-robust "effective" F whose denominator uses the shock-level (AKM-type) variance of the first-stage coefficient — the relevant notion when weak **shocks** are the concern (in the spirit of Montiel Olea & Pflueger 2013, adapted to shift-share).

Usage

```
ssb_first_stage(design)
```

Arguments

design An [ssb_design()] object.

Value

A list (class ‘ssb_first_stage’) with ‘F_standard’, ‘F_effective’, and the first-stage coefficient ‘pi’.

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "share")
ssb_first_stage(d)
```

ssb_loo

Leave-one-sector-out sensitivity

Description

Recomputes the overall estimate dropping each of the top sectors (by IRotemberg weightl) one at a time, to see whether identification hinges on a single shock.

Usage

```
ssb_loo(
  design,
  top = 5,
  se = c("none", "iid", "ehw", "cluster", "akm", "akm0"),
  level = 0.95
)
```

Arguments

design	An [ssb_design()] object.
top	Number of top-weight sectors to leave out in turn.
se	Standard-error method for a confidence interval on each leave-one-out estimate: "none" (default; point estimates only, the original behaviour) or one of "iid", "ehw", "cluster", "akm", "akm0" (each re-estimated on the reduced design via [ssb_estimate()]). With a CI you can read whether the estimate still excludes 0 after dropping the most influential shock; [ssb_plot_loo()] then draws the intervals.
level	Confidence level for the interval when 'se' is not "none".

Value

A 'data.frame' with the dropped 'sector', its 'alpha', and the 'beta_drop' obtained without it (plus the full-sample 'beta_hat' attribute). When 'se' is not "none" it also has 'conf.low'/'conf.high' columns and 'se_method'/'level' attributes.

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "share")
ssb_loo(d, top = 5)
```

ssb_overid

*Overidentification / cross-instrument homogeneity test***Description**

Treats each sector's share as a separate instrument and tests whether the just-identified estimates $\hat{\beta}_n$ are mutually consistent, using a precision-weighted Cochran's Q statistic $Q = \sum_n (\hat{\beta}_n - \bar{\beta})^2 / \widehat{\text{Var}}(\hat{\beta}_n)$ referred to a χ^2_{K-1} distribution. Rejection points to a failure of shares/shocks exogeneity ****or**** to treatment-effect heterogeneity across instruments (Goldsmith-Pinkham, Sorkin & Swift 2020). Very weak instruments are down-weighted automatically; use 'min_F' to drop near-dead instruments entirely.

Usage

```
ssb_overid(design, min_F = 0)
```

Arguments

design	An [ssb_design()] object.
min_F	Drop instruments whose own first-stage F is below this.

Details

****Caveat.**** The $\hat{\beta}_n$ are estimated from the *same* sample and are therefore mutually correlated; the χ^2_{K-1} reference treats them as independent and ignores that covariance. Read the p-value as a heuristic screen for gross cross-instrument disagreement, not as a formal overidentification test — for the latter, use a J-type test with an estimator robust to many instruments (e.g. the HFUL-based test in Goldsmith-Pinkham, Sorkin & Swift 2020).

Value

A list (class 'ssb_overid') with 'Q', 'df', 'p', 'l2', 'beta_bar', 'n_instruments', 'n_dropped'.

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "share")
ssb_overid(d)
```

ssb_pipeline

*Run the full shift-share analysis pipeline***Description**

Given a design, runs the estimation and the route-appropriate battery of diagnostics in one call, dispatching on ‘design\$exogenous’:

- **share** (Goldsmith-Pinkham, Sorkin & Swift 2020): Rotemberg-weight decomposition, leave-one-out sensitivity, and — if ‘covariates’ are supplied — a share-balance check; a pre-trend check if ‘pre_y’ is supplied.
- **shift** (Borusyak, Hull & Jaravel 2022): effective-shock / exposure-concentration summary, leave-one-out sensitivity, and the shock-balance hook.

Estimation always reports the full SE panel (naive / EHW / cluster / AKM / AKM0). The point estimate and first-stage F are common to both routes.

Usage

```
ssb_pipeline(
  design,
  covariates = NULL,
  pre_y = NULL,
  placebo_y = NULL,
  shock_covariates = NULL,
  top = 5,
  level = 0.95
)
```

Arguments

design	An [ssb_design()] object.
covariates	Optional observables for the share-balance check (share route).
pre_y	Optional pre-period outcome for [ssb_pretrend()].
placebo_y	Optional placebo outcome for [ssb_placebo()].
shock_covariates	Optional shock-level characteristics (a data.frame keyed by sector) for [ssb_shock_balance()] on the shift route.
top	Number of top-weight sectors for the sensitivity diagnostics.
level	Confidence level.

Value

An ‘ssb_result’ list with ‘estimate’, ‘route’, and route-specific diagnostic elements. ‘autoplot()’ returns the headline figure.

ssb_placebo	<i>Placebo-outcome test</i>
-------------	-----------------------------

Description

Runs the *same* shift-share IV but on an outcome that the treatment should not move (a placebo). A coefficient far from zero signals that the design is picking up something other than the intended channel. This is distinct from `[ssb_pretrend()]`, which regresses a *pre-period* outcome on the instrument (reduced form) to look for differential pre-trends.

Usage

```
ssb_placebo(
  design,
  placebo_y,
  methods = c("ehw", "cluster", "akm"),
  level = 0.95
)
```

Arguments

<code>design</code>	An <code>[ssb_design()]</code> object.
<code>placebo_y</code>	Column name of the placebo outcome in <code>'data'</code> .
<code>methods</code>	Standard-error methods (see <code>[ssb_estimate()]</code>).
<code>level</code>	Confidence level.

Value

An `'ssb_estimate'` for the placebo outcome.

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
sim$data$y_placebo <- stats::rnorm(nrow(sim$data)) # an unrelated outcome
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "share")
ssb_placebo(d, placebo_y = "y_placebo")
```

ssb_plot_ci

Plot the confidence-interval comparison

Description

Draws the (identical) point estimate with each method's confidence interval, making the practical consequences of the inference method immediately visible — the naive/EHW intervals are typically far too narrow relative to the exposure-robust AKM / AKM0 intervals. The comparison is of **intervals**: AKM0 is defined as an interval directly (and can be asymmetric), so it is the interval, not a standard error, that is the object of interest.

Usage

```
ssb_plot_ci(x, title = NULL, ...)
```

Arguments

<code>x</code>	An 'ssb_estimate' object (from [ssb_estimate()]).
<code>title</code>	Optional plot title.
<code>...</code>	Ignored.

Value

A 'ggplot' object.

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "share")
ssb_plot_ci(ssb_estimate(d))
```

ssb_plot_loo

Leave-one-out sensitivity plot

Description

Plots the shift-share estimate re-computed with each top-weight shock dropped (see [ssb_loo()]) against the full estimate (dashed line), so a result that hinges on a single shock is obvious.

Usage

```
ssb_plot_loo(x, title = NULL, ...)
```

Arguments

x	An [ssb_loo()] object.
title	Optional plot title.
...	Unused.

Value

A **ggplot2** object.

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "share")
ssb_plot_loo(ssb_loo(d, top = 5))
```

ssb_plot_overid	<i>Overidentification dispersion plot</i>
-----------------	---

Description

Forest plot of the just-identified estimates $\hat{\beta}_k$ (one per instrument) with confidence intervals, ordered by size, against the precision-weighted mean (dashed line). Wide, mutually inconsistent estimates signal a failure of the exogeneity assumption or treatment-effect heterogeneity (see [ssb_overid()]). Point size is the first-stage F; the axis is trimmed to the bulk since weak instruments have very wide intervals.

Usage

```
ssb_plot_overid(x, level = 0.95, xlim = NULL, title = NULL, ...)
```

Arguments

x	An [ssb_overid()] object.
level	Confidence level for the per-instrument intervals.
xlim	Optional ‘c(lo, hi)’ for the horizontal axis. By default the axis is trimmed to the bulk of the estimates because weak single-share instruments have very wide intervals; widen it here to show more of them.
title	Optional plot title.
...	Unused.

Value

A **ggplot2** object.

ssb_plot_ri	<i>Randomization-inference plot</i>
-------------	-------------------------------------

Description

Histogram of the permuted-shock (placebo) estimates from [ssb_ri()], with the observed estimate marked; the RI p-value is where the observed value falls in this null distribution.

Usage

```
ssb_plot_ri(x, bins = 30, title = NULL, ...)
```

Arguments

x	An [ssb_ri()] object.
bins	Number of histogram bins.
title	Optional plot title.
...	Unused.

Value

A **ggplot2** object.

ssb_plot_rottemberg	<i>Plot Rotemberg weights (canonical GPSS figure)</i>
---------------------	---

Description

Reproduces Figure 1 of Goldsmith-Pinkham, Sorkin and Swift (2020): each sector-cell is a bubble at its first-stage F-statistic (x) and just-identified estimate $\hat{\beta}_n$ (y); bubble area is proportional to the absolute Rotemberg weight; positive-weight cells are blue open circles and negative ones amber open diamonds; the dashed horizontal line marks the overall estimate $\hat{\beta}$.

Usage

```
ssb_plot_rottemberg(x, max_size = 12, label_top = 0, title = NULL, ...)
```

Arguments

x	An ‘ssb_rottemberg’ object (from [ssb_rottemberg()]).
max_size	Maximum bubble size.
label_top	If > 0, label this many top-weight sectors.
title	Optional plot title.
...	Ignored.

Value

A ‘ggplot’ object.

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "share")
ssb_plot_rottemberg(ssb_rottemberg(d))
```

ssb_plot_shocks	<i>Exposure-concentration (Lorenz) plot</i>
-----------------	---

Description

Lorenz curve of the shock exposure weights from [ssb_shock_summary()]: the further the curve bows below the 45-degree line, the more the identifying variation is concentrated in a few shocks. The effective number of shocks and the HHI are shown in the subtitle.

Usage

```
ssb_plot_shocks(x, title = NULL, ...)
```

Arguments

x	An [ssb_shock_summary()] (‘ssb_shocks’) object.
title	Optional plot title.
...	Unused.

Value

A **ggplot2** object.

ssb_pretrend	<i>Pre-trend test</i>
--------------	-----------------------

Description

Reduced-form regression of a pre-period outcome on the constructed instrument (controls partialled out). A coefficient far from zero indicates that exposure predicts differential pre-trends — a threat to identification. This is distinct from [ssb_placebo()], which runs the **full IV** on a placebo outcome; pre-trends ask whether exposure predicts the outcome **before** the shocks, placebo asks whether the design moves an outcome it should not.

Usage

```
ssb_pretrend(design, pre_y, level = 0.95)
```

Arguments

<code>design</code>	An <code>[ssb_design()]</code> object.
<code>pre_y</code>	Column name of the pre-period outcome (or pre-period change).
<code>level</code>	Confidence level.

Details

Because the regressor is itself a shift-share variable, EHW / cluster standard errors are subject to exactly the over-rejection documented by Adao, Kolesar & Morales (2019): residuals are correlated across units with similar exposure. The test therefore also reports an exposure-robust (AKM-type) standard error that clusters the score at the shock level; treat that one as the headline, especially on the shift route, or spurious "pre-trends" will appear too often.

Value

A list (class `'ssb_pretrend'`) with the reduced-form coefficient, EHW / cluster / exposure-robust (AKM) standard errors, the corresponding p-values (`'p_ehw'`, `'p_akm'`), and intervals (`'conf.low'`/`'conf.high'` use the EHW SE; `'conf.low_akm'`/`'conf.high_akm'` the exposure-robust SE).

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
sim$data$y_pre <- stats::rnorm(nrow(sim$data)) # a pre-period outcome
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "share")
ssb_pretrend(d, pre_y = "y_pre")
```

ssb_recenter

Recenter the shocks (Borusyak & Hull)

Description

Recentering removes the expected instrument implied by the shock-assignment process, so identification comes only from deviations of shocks from their (conditional) mean. Two methods:

- `"demean"` (default): subtract the single exposure-weighted mean shock $\bar{g}$. Leaves the point estimate unchanged but makes the identifying variation explicit.
- `"permute"`: subtract the **block-specific simple average** shock, i.e. recenter within exchangeability groups. Under uniform within-block permutation every cell in a block is equally likely to receive each of the block's shocks, so $E[g_n]$ is the unweighted within-block mean; subtracting it gives the expectation of the instrument under that assignment process (Borusyak & Hull), computed analytically. With no `'block'` this recenters by the grand unweighted mean.

For randomization-inference p-values based on the same permutation idea, see `[ssb_ri()]`.

Usage

```
ssb_recenter(design, method = c("demean", "permute"), block = NULL, ...)
```

Arguments

design	An [ssb_design()] object.
method	"demean" or "permute".
block	Exchangeability blocks for "permute": a column name in the shocks table, or a vector of length equal to the number of shock-cells.
...	Reserved.

Value

A new [ssb_design()] with recentered shocks/instrument (carries a "recentered" attribute).

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "shift")
ssb_recenter(d, method = "demean")
```

ssb_ri

Randomization-inference (placebo-shock) test

Description

Re-draws the shocks by permutation (optionally within exchangeability 'block's) and reports where the observed statistic falls in the resulting placebo distribution, in the spirit of Adao-Kolesar-Morales (2019) and Borusyak & Hull.

Usage

```
ssb_ri(design, R = 999, block = NULL, null = 0, seed = NULL)
```

Arguments

design	An [ssb_design()] object.
R	Number of permutation draws.
block	Optional exchangeability blocks for shocks: a column name in the shocks table, or a vector of length equal to the number of shock-cells. Shocks are permuted only within blocks. In sector x period panels you almost always want blocks that separate periods, so shocks are not permuted across time.
null	The null value β_0 of the coefficient (default 0).
seed	Optional RNG seed.

Details

The statistic is Anderson-Rubin-style: the reduced-form coefficient of $y - \beta_0 x$ on the reconstructed instrument, with $\beta_0 = \text{'null'}$. Under the constant-effects null $\beta = \beta_0$ (plus the exclusion restriction), $y - \beta_0 x$ does not respond to how the shocks are assigned, so the permutation distribution of this statistic is **exact** given the exchangeability encoded in *'block'*. Permuting the IV ratio itself (holding the observed treatment fixed) would **not** be exact — the treatment also responds to the shocks through the first stage, and placebo draws with weak first stages give the ratio very heavy tails — so this function does not do that.

Value

A list (class *'ssb_ri'*) with the IV point estimate *'beta'*, the observed Anderson-Rubin *'statistic'*, *'null'*, *'p_value'*, *'R'*, and the vector *'perm'* of placebo statistics.

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "share")
ssb_ri(d, R = 199, seed = 1)
```

ssb_rotemberg

Rotemberg weights for a Bartik instrument

Description

Decomposes the shift-share 2SLS estimate into a weighted sum of the just-identified estimates that use each sector's share as a single instrument, following Goldsmith-Pinkham, Sorkin and Swift (2020):

$$\hat{\beta} = \sum_n \hat{\alpha}_n \hat{\beta}_n, \quad \hat{\alpha}_n = \frac{g_n \tilde{s}'_n \tilde{x}}{\sum_{n'} g_{n'} \tilde{s}'_{n'} \tilde{x}},$$

where tildes denote residualisation on the controls (and, in panels, sector-cells are sector $\times$ period pairs). The weights $\hat{\alpha}_n$ sum to one and measure the sensitivity of $\hat{\beta}$ to misspecification of each sector's instrument; a small number of large weights is a warning sign. Unlike Goodman-Bacon weights, negative Rotemberg weights are not automatically problematic.

Usage

```
ssb_rotemberg(design)
```

Arguments

design An [ssb_design()] object.

Value

A ‘data.frame’ of class ‘ssb_rottemberg’, one row per sector-cell, with columns ‘sector’, ‘g’ (shock), ‘alpha’ (Rotemberg weight), ‘beta’ (just-identified estimate), ‘F’ (first-stage F of that instrument), and ‘sign’. Carries the overall estimate ‘beta_hat’ as an attribute. Pass it to [ssb_plot_rottemberg()] for the canonical figure.

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "share")
ssb_rottemberg(d)
```

ssb_share_balance	<i>Share balance (exogenous-shares route)</i>
-------------------	---

Description

For the top-exposure sectors, regresses each sector’s share on observable unit characteristics to see how strongly exposure correlates with observables — the key credibility check when identification comes from the shares.

Usage

```
ssb_share_balance(design, covariates, top = 5)
```

Arguments

design	An [ssb_design()] object.
covariates	Character vector of observable columns in ‘data’.
top	Number of top-exposure sectors to test.

Value

A ‘data.frame’ of slope coefficients and (robust) t-statistics of each covariate in the share regression, one block per tested sector.

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "share")
ssb_share_balance(d, covariates = "w1", top = 3)
```

ssb_shock_balance	<i>Shock-level balance test</i>
-------------------	---------------------------------

Description

Tests the identifying assumption of the shocks route — that shocks are as-good-as-randomly assigned — by regressing the shocks on pre-determined shock-level characteristics, weighted by exposure (Borusyak, Hull & Jaravel 2022). Coefficients near zero and a non-significant joint test support shock exogeneity.

Usage

```
ssb_shock_balance(design, shock_covariates, weight = TRUE)
```

Arguments

design	An [ssb_design()] object.
shock_covariates	A ‘data.frame’ keyed by ‘sector’ (and ‘time’ for panels) holding the shock-level characteristics to test.
weight	If ‘TRUE’ (default) weight by exposure s_n ; else unweighted.

Value

A list (class ‘ssb_shock_balance’) with a coefficient table and the joint Wald test that the characteristics are unrelated to the shocks.

ssb_shock_iv	<i>Shock-level IV estimate</i>
--------------	--------------------------------

Description

Runs the exposure-weighted IV at the shock level (see [ssb_aggregate()]). The point estimate equals the location-level shift-share estimate; the shock-level heteroskedasticity- or cluster-robust standard error here is the natural shock-level analogue of the AKM exposure-robust SE.

Usage

```
ssb_shock_iv(design, cluster = NULL, level = 0.95)
```

Arguments

design	An [ssb_design()] object.
cluster	Optional vector (length = number of shock-cells) grouping shocks into clusters for the shock-level SE.
level	Confidence level for the reported interval.

Value

A one-row ‘data.frame’ (class ‘ssb_shock_iv’) with ‘estimate’, ‘std.error’, ‘conf.low’, ‘conf.high’.

ssb_shock_summary	<i>Shock summary: effective number of shocks and exposure concentration</i>
-------------------	---

Description

Reports the Borusyak-Hull-Jaravel (2022) exposure-concentration diagnostics for the shocks route: the average exposure (importance) weight of each shock, its Herfindahl index, and the *effective number of shocks* $1 / \sum_n \bar{s}_n^2$. Few effective shocks undermine the large-n asymptotics that justify the shocks-exogeneity approach.

Usage

```
ssb_shock_summary(design)
```

Arguments

design An [ssb_design()] object.

Value

A list with ‘effective_shocks’, ‘hhi’, ‘n_shocks’, and a ‘data.frame’ ‘weights’ of per-shock importance weights (descending). Class ‘ssb_shocks’.

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "shift")
ssb_shock_summary(d)
```

ssb_table_image	<i>Render a result table as an image (PNG or PDF)</i>
-----------------	---

Description

‘plot()’ methods that draw the same booktabs-style table [format()] prints, as a standalone image — the sibling of [plot.ssb_rotemberg()]. Pass ‘file=’ to write a ‘.png’ (default) or ‘.pdf’; without ‘file’ the table is drawn on the current graphics device. For LaTeX/Markdown source instead of an image, use ‘format(x, "latex”)’ / ‘format(x, "markdown”)’.

Usage

```
## S3 method for class 'ssb_estimate'
plot(x, file = NULL, width = NULL, height = NULL, res = 200, digits = 3, ...)

## S3 method for class 'ssb_weight_summary'
plot(x, file = NULL, width = NULL, height = NULL, res = 200, digits = 3, ...)

## S3 method for class 'ssb_overid'
plot(x, file = NULL, width = NULL, height = NULL, res = 200, digits = 3, ...)

## S3 method for class 'ssb_loo'
plot(x, file = NULL, width = NULL, height = NULL, res = 200, digits = 3, ...)

## S3 method for class 'ssb_drop_top'
plot(x, file = NULL, width = NULL, height = NULL, res = 200, digits = 3, ...)
```

Arguments

x	A result object: an [ssb_estimate()] (also [ssb_placebo()], [ssb_weight_summary()], [ssb_overid()], [ssb_loo()], or [ssb_drop_top()]).
file	Output path ('.png' or '.pdf'); 'NULL' draws on the current device.
width, height	Image size in inches (auto-sized when 'NULL').
res	PNG resolution in dpi.
digits	Number of decimal places.
...	Unused.

Value

The 'file' path, invisibly.

ssb_weight_summary	<i>Rotemberg-weight summary and correlations (GPSS diagnostic table)</i>
--------------------	--

Description

Summarises the Rotemberg-weight diagnostic in the spirit of Goldsmith-Pinkham, Sorkin & Swift (2020): the top-weight shocks, the largest single weight, the correlation of the weights with the just-identified estimates and first-stage F, and — if 'covariates' are supplied — the correlation between each shock's Rotemberg weight and its exposure-weighted average of unit observables (do high-weight shocks load on systematically different places?).

Usage

```
ssb_weight_summary(design, covariates = NULL, top = 5)
```

Arguments

design	An [ssb_design()] object.
covariates	Optional unit-level observable columns in 'data'.
top	Number of top-weight shocks to display.

Value

A list (class 'ssb_weight_summary').

Examples

```
sim <- ssb_simulate(n_loc = 80, n_sec = 10, seed = 1)
d <- ssb_design(sim$data, sim$shares, sim$shocks, exogenous = "share")
ssb_weight_summary(d, covariates = "w1")
```

ssbartik

One-call shift-share analysis

Description

Convenience wrapper that builds an [ssb_design()] from raw pieces and runs [ssb_pipeline()] — the "give me everything" entry point. Specify the identification route with 'exogenous' and the rest flows through to diagnostics and plots.

Usage

```
ssbartik(
  data,
  shares,
  shocks,
  y = "y",
  x = "x",
  location = "location",
  sector = "sector",
  time = NULL,
  controls = NULL,
  weights = NULL,
  cluster = NULL,
  share_col = "share",
  shock_col = "shock",
  exogenous = c("shift", "share"),
  covariates = NULL,
  pre_y = NULL,
  placebo_y = NULL,
  shock_covariates = NULL,
  top = 5,
  level = 0.95
)
```

Arguments

<code>data</code>	A unit-level 'data.frame': one row per location (or location-period). Must contain 'y', 'x', 'location', and any 'controls', 'weights', 'cluster' columns referenced below.
<code>shares</code>	A long 'data.frame' of exposure shares with columns 'location', 'sector', the share column ('share_col'), and 'time' for panels.
<code>shocks</code>	A 'data.frame' of shocks with columns 'sector', the shock column ('shock_col'), and 'time' for panels.
<code>y, x</code> <code>location, sector</code>	Column names (strings) of the outcome and endogenous treatment. Column names of the unit and sector identifiers.
<code>time</code>	Optional column name of a period identifier (present in 'data', 'shares' and 'shocks') for panel designs.
<code>controls</code>	Optional character vector of control columns in 'data'. Numeric columns enter linearly; factor or character columns are expanded into dummies, so period or region fixed effects can be supplied as factors (in panel shift-share designs, period fixed effects are usually essential — shocks should be compared within periods).
<code>weights</code>	Optional column name of regression weights in 'data'.
<code>cluster</code>	Optional column name of a clustering variable in 'data'.
<code>share_col</code>	Name of the exposure-share column in 'shares' (default "share").
<code>shock_col</code>	Name of the shock (shift) column in 'shocks' (default "shock").
<code>exogenous</code>	Which identification route to emphasise downstream: "shift" (shocks) or "share" (shares). "shock"/"shares" are accepted aliases.
<code>covariates, pre_y, placebo_y, shock_covariates, top, level</code>	Passed to [ssb_pipeline()].

Value

An 'ssb_result' object.

Examples

```
# Bring your own data; this is a small synthetic design for illustration.
set.seed(1)
n_loc <- 60L; n_sec <- 8L
shares <- expand.grid(location = seq_len(n_loc), sector = seq_len(n_sec))
shares$share <- stats::runif(nrow(shares))
tot <- tapply(shares$share, shares$location, sum)
shares$share <- shares$share / tot[as.character(shares$location)]
shocks <- data.frame(sector = seq_len(n_sec), shock = stats::rnorm(n_sec))
Z <- tapply(shares$share, list(shares$location, shares$sector), sum)
Z[is.na(Z)] <- 0
inst <- as.numeric(Z %*% shocks$shock)
dat <- data.frame(location = seq_len(n_loc),
  x = 4 * inst + stats::rnorm(n_loc, sd = 0.3))
```

```
dat$y <- 1.2 * dat$x + stats::rnorm(n_loc, sd = 0.3)
res <- ssbartik(dat, shares, shocks, exogenous = "share")
res

autoplot(res)                # headline Rotemberg figure
autoplot(res$estimate)       # CI comparison
```

Index

`format.ssb_drop_top`, 3
`format.ssb_estimate`, 3
`format.ssb_loo`, 4
`format.ssb_overid`, 5
`format.ssb_rotemberg`, 6
`format.ssb_shock_balance`, 7
`format.ssb_shocks`, 7
`format.ssb_weight_summary`, 8

`plot.ssb_drop_top(ssb_table_image)`, 29
`plot.ssb_estimate(ssb_table_image)`, 29
`plot.ssb_loo(ssb_table_image)`, 29
`plot.ssb_overid(ssb_table_image)`, 29
`plot.ssb_rotemberg`, 9
`plot.ssb_weight_summary`
 (`ssb_table_image`), 29

`ssb_aggregate`, 10
`ssb_design`, 11
`ssb_drop_top`, 12
`ssb_equivalence`, 13
`ssb_estimate`, 14
`ssb_first_stage`, 15
`ssb_loo`, 16
`ssb_overid`, 17
`ssb_pipeline`, 18
`ssb_placebo`, 19
`ssb_plot_ci`, 20
`ssb_plot_loo`, 20
`ssb_plot_overid`, 21
`ssb_plot_ri`, 22
`ssb_plot_rotemberg`, 22
`ssb_plot_shocks`, 23
`ssb_pretrend`, 23
`ssb_recenter`, 24
`ssb_ri`, 25
`ssb_rotemberg`, 26
`ssb_share_balance`, 27
`ssb_shock_balance`, 28
`ssb_shock_iv`, 28
`ssb_shock_summary`, 29
`ssb_table_image`, 29
`ssb_weight_summary`, 30
`ssbartik`, 31