Package 'spliv'

July 29, 2026

Type Package

Title Patterned Sensitivity Analysis for IV with Fixed Effects

Version 0.1.1

Description Patterned sensitivity analysis for instrumental-variables designs with fixed effects or other residualization steps. The package provides uniform Conley-style sensitivity as a baseline, researcher-specified direct-effect patterns, sensitivity paths and tipping points, and optional confirmatory Beyond Plausibly Exogenous diagnostics.

License GPL-3

Encoding UTF-8

Depends R (>= 4.1)

Imports stats, graphics, fixest

Suggests lfe, testthat (>= 3.0.0), ggplot2, knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

URL https://koren6684.github.io/spliv/, https://github.com/koren6684/spliv

BugReports https://github.com/koren6684/spliv/issues

RoxygenNote 7.3.3

NeedsCompilation no

Author Ore Koren [aut, cre]

Maintainer Ore Koren <okoren@iu.edu>

Repository CRAN

Date/Publication 2026-07-29 16:50:02 UTC

Contents

`bpe_design` *Create a Confirmatory BPE Design Object*

Description

Creates a researcher-specified design object for confirmatory BPE. The subset must be justified outside the outcome analysis and should represent a theoretically motivated instrument-inactive subset.

Usage

```
bpe_design(
  name,
  subset,
  rationale,
  variables_used = NULL,
  subset_type = NULL,
  pre_specified = TRUE,
  transportability_rationale = NULL,
  notes = NULL
)
```

Arguments

name	Short design name.
subset	Subset specification. Accepts a one-sided formula, a function(data), a logical vector of length nrow(data), or a character string naming a logical column in data.
rationale	Non-empty substantive justification for the subset.
variables_used	Optional character vector describing the design variables used to define the subset. This is required for function-based subsets unless the package can safely infer the variables.
subset_type	Optional short label such as "theory_defined" or "design_based".
pre_specified	Logical indicator for confirmatory use. Confirmatory BPE requires TRUE.
transportability_rationale	Optional description of why the direct effect learned in the subset may be informative for the target sample.
notes	Optional free-form notes.

Value

An object of class "spliv_bpe_design".

Examples

```
design <- bpe_design(
  "Inactive subset", ~ inactive,
  rationale = "The treatment channel is absent in this subset.",
  variables_used = "inactive", pre_specified = TRUE
)
bpe_eval_subset(design, data.frame(inactive = c(TRUE, FALSE)))
```

---

bpe_eval_subset *Evaluate a Confirmatory BPE Subset*

Description

Evaluates a bpe_design() object on a data frame and returns the resulting logical indicator.

Usage

```
bpe_eval_subset(design, data, max_na_share = 0.05)
```

Arguments

design	A bpe_design() object.
data	Data frame used for evaluation.
max_na_share	Maximum allowable share of NA values in the subset indicator before evaluation fails. The default is 0.05.

Value

A logical vector of length `nrow(data)`.

Examples

```
design <- bpe_design("Inactive", ~ inactive,
  rationale = "The treatment channel is absent.")
bpe_eval_subset(design, data.frame(inactive = c(TRUE, FALSE)))
```

`bpe_explore_subsets` *Explore Candidate BPE Subsets*

Description

Explores user-supplied candidate subset definitions and reports diagnostics for transparent exploratory work. This function is not valid confirmatory BPE by itself. Confirmatory BPE requires a pre-specified `bpe_design()` object supplied to `spliv()` or `bpe_validate_design()`.

Usage

```
bpe_explore_subsets(
  data,
  spec,
  rules = NULL,
  seed = 1,
  min_n_S = NULL,
  max_F_S = NULL,
  min_varZ_S = NULL,
  return_all = TRUE
)
```

Arguments

`data`	Data frame used for estimation.
`spec`	Either an IV formula (`y ~ X | Z`) or a list with at least `formula`, and optional `fe`, `fe_engine`, and `z_names`.
`rules`	Candidate subset definitions. Each candidate may be a one-sided formula evaluated in `data`, a `function(data)` returning a logical vector, a character string naming a logical column in `data`, a logical vector of length `nrow(data)`, or a named list with components `name` and `subset`.
`seed`	Integer seed for deterministic rule evaluation.
`min_n_S`	Optional exploratory screen for subset size.
`max_F_S`	Deprecated exploratory screen for the first-stage F-statistic. The first-stage F-statistic is reported as a diagnostic only.
`min_varZ_S`	Optional exploratory screen for within-subset residualized instrument variance.

`return_all` If `TRUE` (default), returns all candidates and diagnostics. If `FALSE`, returns the first candidate in the user-supplied order. No automatic subset search or confirmatory selection is performed.

Details

Warning: this function is exploratory. It should not be used for confirmatory BPE inference. Confirmatory BPE requires a pre-specified `bpe_design()` object with a substantive rationale.

Value

If `return_all = TRUE`, a data frame with one row per rule and a list column `subset`. If `return_all = FALSE`, a list with the first supplied `subset`, `rule`, and `diagnostics`.

Examples

```
d <- data.frame(y = rnorm(40), x = rnorm(40), z = rnorm(40),
  inactive = rep(c(TRUE, FALSE), each = 20))
suppressWarnings(bpe_explore_subsets(
  d, y ~ x | z, rules = list(inactive = ~ inactive)))
```

`bpe_find_subset` *Deprecated Exploratory BPE Subset Search*

Description

`bpe_find_subset()` is retained for backward compatibility only. Use `bpe_explore_subsets()` for exploratory work and convert any theory-justified subset into a confirmatory `bpe_design()` object before calling `spliv()`.

Usage

```
bpe_find_subset(
  data,
  spec,
  rules = NULL,
  seed = 1,
  min_n_S = NULL,
  max_F_S = NULL,
  min_varZ_S = NULL,
  return_all = TRUE
)
```

Arguments

data	Data frame used for estimation.
spec	Either an IV formula (y ~ X | Z) or a list with at least formula, and optional fe, fe_engine, and z_names.
rules	Candidate subset definitions. Each candidate may be a one-sided formula evaluated in data, a function(data) returning a logical vector, a character string naming a logical column in data, a logical vector of length nrow(data), or a named list with components name and subset.
seed	Integer seed for deterministic rule evaluation.
min_n_S	Optional exploratory screen for subset size.
max_F_S	Deprecated exploratory screen for the first-stage F-statistic. The first-stage F-statistic is reported as a diagnostic only.
min_varZ_S	Optional exploratory screen for within-subset residualized instrument variance.
return_all	If TRUE (default), returns all candidates and diagnostics. If FALSE, returns the first candidate in the user-supplied order. No automatic subset search or confirmatory selection is performed.

Value

See bpe_explore_subsets().

Examples

```
d <- data.frame(y = rnorm(40), x = rnorm(40), z = rnorm(40),
  inactive = rep(c(TRUE, FALSE), each = 20))
suppressWarnings(bpe_find_subset(
  d, y ~ x | z, rules = list(inactive = ~ inactive)))
```

---

bpe_validate_design *Validate a Confirmatory BPE Design*

Description

Runs the confirmatory BPE eligibility diagnostics for a pre-specified design without fitting the final SPLIV model.

Usage

```
bpe_validate_design(
  formula,
  data,
  design,
  fe = NULL,
  fe_engine = c("fixest", "lfe"),
  vcov = c("iid", "hc1", "cluster"),
```

```
  cluster = NULL,
  z_names = NULL,
  bpe_min_n_S = 2000,
  bpe_min_clusters_S = 30,
  bpe_min_varZ_S = 1e-06,
  bpe_equiv_margin,
  bpe_equiv_level = 0.95,
  bpe_transport = c("none", "sampling", "conservative"),
  bpe_transport_kappa = 0,
  bpe_kappa = 1,
  scale_instrument = c("residual_sd", "none")
)
```

Arguments

`formula`	IV formula `y ~ X | Z`.
`data`	Data frame.
`design`	A `bpe_design()` object.
`fe`	Optional one-sided formula of fixed effects.
`fe_engine`	FE demeaning engine, one of `"fixest"` or `"lfe"`.
`vcov`	One of `"iid"`, `"hc1"`, or `"cluster"`.
`cluster`	Cluster ids or one-sided formula when `vcov = "cluster"`.
`z_names`	Optional instrument name for BPE. Confirmatory BPE currently supports exactly one instrument.
`bpe_min_n_S`	Minimum subset size threshold.
`bpe_min_clusters_S`	Minimum number of clusters required in the subset when clustered covariance is used.
`bpe_min_varZ_S`	Minimum residualized instrument variance required in the subset.
`bpe_equiv_margin`	Researcher-specified equivalence margin for the first-stage coefficient. Eligibility is based on the first-stage equivalence interval, not on the first-stage F-statistic.
`bpe_equiv_level`	Confidence level used for the first-stage equivalence interval.
`bpe_transport`	One of `"none"`, `"sampling"`, or `"conservative"`. Transportability is an assumption reflected in the reported covariance; it is not established by the subset itself.
`bpe_transport_kappa`	Non-negative scalar controlling the conservative transport covariance inflation.
`bpe_kappa`	Positive scalar multiplier applied to the transported BPE covariance before it is embedded into the LTZ prior.
`scale_instrument`	One of `"residual_sd"` or `"none"`.

Value

An object of class "spliv_bpe_validation" containing design metadata, subset diagnostics, first-stage equivalence diagnostics, reduced- form direct-effect estimates, and covariance components for confirmatory BPE.

Examples

```
set.seed(2)
d <- data.frame(
  y = rnorm(80), x = rnorm(80), z = rnorm(80),
  inactive = rep(c(TRUE, FALSE), each = 40)
)
design <- bpe_design("Inactive", ~ inactive,
  rationale = "The treatment channel is absent.")
bpe_validate_design(y ~ x | z, d, design,
  bpe_min_n_S = 20, bpe_equiv_margin = 1)
```

---

demean_fixest *Demean with fixest*

Description

Demeans vectors/matrices by high-dimensional fixed effects using `fixest::demean`.

Usage

```
demean_fixest(y, X, Z, W = NULL, fe_fml, data)
```

Arguments

`y`	Outcome vector.
`X`	Regressor matrix.
`Z`	Instrument matrix.
`W`	Optional controls matrix.
`fe_fml`	One-sided FE formula.
`data`	Data frame aligned with the rows of `y`, `X`, `Z`, `W`.

Value

List with demeaned `y`, `X`, `Z`, `W`, and FE frame.

Examples

```
d <- data.frame(g = factor(rep(1:2, each = 3)))
y <- 1:6; X <- matrix(rnorm(6), 6, 1); Z <- matrix(rnorm(6), 6, 1)
demean_fixest(y, X, Z, fe_fml = ~ g, data = d)
```

demean_lfe *Demean with lfe*

Description

Demeans vectors/matrices by high-dimensional fixed effects using `lfe::demeanlist`.

Usage

```
demean_lfe(y, X, Z, W = NULL, fe_list)
```

Arguments

`y`	Outcome vector.
`X`	Regressor matrix.
`Z`	Instrument matrix.
`W`	Optional controls matrix.
`fe_list`	List/data.frame of FE ids.

Value

List with demeaned `y`, `X`, `Z`, `W`.

Examples

```
d <- data.frame(g = factor(rep(1:2, each = 3)))
y <- 1:6; X <- matrix(rnorm(6), 6, 1); Z <- matrix(rnorm(6), 6, 1)
demean_lfe(y, X, Z, fe_list = d["g"])
```

embed_prior_into_full_Z *Embed Prior into Full Instrument Space*

Description

Expands a prior estimated on a subset of instruments into the full instrument vector implied by `y ~ X | Z`.

Usage

```
embed_prior_into_full_Z(formula, data, z_names, mu_hat, omega_hat)
```

Arguments

formula	IV formula y ~ X | Z.
data	Data frame.
z_names	Instrument names with prior moments.
mu_hat	Prior means for z_names.
omega_hat	Prior covariance for z_names.

Value

List with full-length mu, Omega, and instrument names.

Examples

```
d <- data.frame(y = rnorm(30), x = rnorm(30), z1 = rnorm(30), z2 = rnorm(30))
embed_prior_into_full_Z(y ~ x | z1 + z2, d,
  z_names = "z1", mu_hat = 0, omega_hat = matrix(0.1, 1, 1))
```

---

estimate_gamma_zero_first_stage

Legacy Exploratory BPE Prior Helper

Description

estimate_gamma_zero_first_stage() is retained as a legacy exploratory helper. It is not sufficient for confirmatory BPE inference. Confirmatory BPE requires bpe_design(), bpe_validate_design(), and spliv(method = "bpe", ...).

Usage

```
estimate_gamma_zero_first_stage(
  data,
  y_name,
  z_names,
  controls = character(0),
  subset,
  fe = NULL
)
```

Arguments

data	Data frame.
y_name	Outcome variable name.
z_names	Instrument names whose direct effects are estimated.
controls	Optional character vector of additional controls.

subset	Legacy subset specification. Accepts a bpe_design() object, a one-sided formula, a function(data), a logical vector, or a character string naming a logical column in data.
fe	Optional one-sided FE formula. If supplied, uses fixest::feols.

Value

List with mu_hat, omega_hat, and diagnostics.

Examples

```
d <- data.frame(y = rnorm(40), z = rnorm(40), inactive = rep(c(TRUE, FALSE), each = 20))
suppressWarnings(estimate_gamma_zero_first_stage(
  d, y_name = "y", z_names = "z", subset = ~ inactive))
```

---

iv_inst_names *Instrument Names for Parsed IV Formula*

Description

Returns the instrument-matrix column order used internally.

Usage

```
iv_inst_names(formula, data)
```

Arguments

formula	IV formula y ~ X | Z.
data	Data frame.

Value

Character vector of instrument names.

Examples

```
d <- data.frame(y = 1:4, x = 1:4, z = 4:1)
iv_inst_names(y ~ x | z, d)
```

`plot_sp_sensitivity` *Plot Patterned Sensitivity Output or a Fitted Object*

Description

Plot Patterned Sensitivity Output or a Fitted Object

Usage

```
plot_sp_sensitivity(
  df_plot,
  ylab = "Effect (beta)",
  main = "Plausibly Exogenous IV Sensitivity",
  ...
)

plot_conley_sensitivity(
  df_plot,
  ylab = "Effect (beta)",
  main = "Plausibly Exogenous IV Sensitivity",
  ...
)
```

Arguments

`df_plot`	Data frame returned by a sensitivity helper or `plausexog_fit` object.
`ylab`	Y-axis label.
`main`	Plot title.
`...`	Unused.

Value

Invisibly returns the plotted input.

Examples

```
set.seed(7)
d <- data.frame(y = rnorm(80), x = rnorm(80), z = rnorm(80))
p <- spliv_sensitivity_path(y ~ x | z, d, method = "uci", delta_grid = c(0, 0.1))
plot_sp_sensitivity(p, term = "x")
```

spliv *Patterned Sensitivity Analysis for Plausibly Exogenous IV*

Description

Main estimator for patterned sensitivity analysis of exclusion violations in fixed-effect or residualized IV designs.

Usage

```
plausexog(
  formula,
  data,
  fe = NULL,
  fe_engine = c("fixest", "lfe"),
  vcov = c("iid", "hc1", "cluster"),
  cluster = NULL,
  method = c("ltz", "uci", "bpe"),
  prior = NULL,
  delta = NULL,
  violation_pattern = NULL,
  bpe = FALSE,
  bpe_design = NULL,
  bpe_spec = list(design = NULL, subset = NULL, z_names = NULL),
  bpe_kappa = 1,
  bpe_omega = NULL,
  bpe_min_n_S = 2000,
  bpe_min_clusters_S = 30,
  bpe_max_F_S = NULL,
  bpe_min_varZ_S = 1e-06,
  bpe_equiv_margin = NULL,
  bpe_equiv_level = 0.95,
  bpe_transport = c("sampling", "none", "conservative"),
  bpe_transport_kappa = 0,
  bpe_not_applicable = c("na", "error"),
  scale_instrument = c("residual_sd", "none"),
  grid = list(),
  ...
)

plausexog_iv(...)

spliv(
  formula,
  data,
  fe = NULL,
  fe_engine = c("fixest", "lfe"),
```

```
  vcov = c("iid", "hc1", "cluster"),
  cluster = NULL,
  method = c("ltz", "uci", "bpe"),
  prior = NULL,
  delta = NULL,
  violation_pattern = NULL,
  bpe = FALSE,
  bpe_design = NULL,
  bpe_spec = list(design = NULL, subset = NULL, z_names = NULL),
  bpe_kappa = 1,
  bpe_omega = NULL,
  bpe_min_n_S = 2000,
  bpe_min_clusters_S = 30,
  bpe_max_F_S = NULL,
  bpe_min_varZ_S = 1e-06,
  bpe_equiv_margin = NULL,
  bpe_equiv_level = 0.95,
  bpe_transport = c("sampling", "none", "conservative"),
  bpe_transport_kappa = 0,
  bpe_not_applicable = c("na", "error"),
  scale_instrument = c("residual_sd", "none"),
  grid = list(),
  ...
)
```

Arguments

`formula`	IV formula `y ~ X | Z`.
`data`	Data frame.
`fe`	Optional one-sided formula of fixed effects.
`fe_engine`	FE demeaning engine, one of `"fixest"` or `"lfe"`.
`vcov`	One of `"iid"`, `"hc1"`, or `"cluster"`.
`cluster`	Cluster ids or one-sided formula, required when `vcov = "cluster"`.
`method`	One of `"ltz"`, `"uci"`, or `"bpe"`.
`prior`	Optional prior list with `mu` and `Omega` (or `omega`) for LTZ. When `violation_pattern` is supplied, patterned LTZ currently requires a scalar prior over the pattern coefficient.
`delta`	Optional non-negative scalar sensitivity magnitude. With `method = "uci"`, `delta` implies theta bounds `[-delta, +delta]` unless explicit bounds are supplied in `grid`. With `method = "ltz"` and no explicit `prior`, `delta` induces a zero-mean normal LTZ prior.
`violation_pattern`	Optional `spliv_pattern()` object describing how the direct effect of the instrument may vary across observations. If omitted, LTZ/UCI retain the package's backward-compatible uniform direct-effect behavior. This argument is currently supported for LTZ and UCI, but not for confirmatory BPE.

bpe — Logical; when TRUE, learn prior moments from a confirmatory bpe_design() and run LTZ.

bpe_design — Optional bpe_design() object for confirmatory BPE.

bpe_spec — Optional list. Prefer bpe_spec = list(design = my_design). For backward compatibility, bpe_spec$subset may also be supplied, but it must represent an explicit researcher-supplied subset and should be paired with a non-empty rationale, explicit pre_specified = TRUE, and any relevant metadata such as variables_used or subset_type. Supply exactly one confirmatory subset source: bpe_design, bpe_spec$design, or bpe_spec$subset. Exploratory subset_rule inputs are not accepted for confirmatory estimation.

bpe_kappa — Positive scalar multiplier applied to the confirmatory BPE covariance after transport adjustment.

bpe_omega — Deprecated. Confirmatory BPE now uses the full reduced-form covariance from the subset together with bpe_transport.

bpe_min_n_S — Minimum subset size required for BPE eligibility. Default 2000.

bpe_min_clusters_S — Minimum number of clusters required in subset S when vcov = "cluster". Default 30.

bpe_max_F_S — Deprecated. The first-stage F-statistic is reported for diagnostics only and no longer determines confirmatory BPE eligibility.

bpe_min_varZ_S — Minimum residualized instrument variance required in subset S. Default 1e-6.

bpe_equiv_margin — Researcher-specified first-stage equivalence margin. Confirmatory BPE currently supports exactly one instrument.

bpe_equiv_level — Confidence level for the first-stage equivalence check.

bpe_transport — One of "none", "sampling", or "conservative".

bpe_transport_kappa — Non-negative scalar controlling the conservative transport covariance inflation.

bpe_not_applicable — Behavior when subset diagnostics fail. One of "na" (default) to return NA estimates, or "error" to stop.

scale_instrument — One of "residual_sd" (default) or "none".

grid — List controlling UCI bounds or other tuning parameters. For backward-compatible scalar UCI, if grid$delta is supplied and grid$gmin/grid$gmax are omitted, the package interprets delta as a direct-effect bound of [-delta, +delta] under the chosen scale_instrument. When violation_pattern is supplied, grid$delta instead refers to theta bounds over the pattern-scaled direct effect.

... — Reserved.

Details

spliv implements patterned sensitivity analysis for exclusion violations in IV designs with fixed effects or other residualization steps. Researchers can supply a theoretically motivated spliv_pattern()

object to specify where direct effects of an instrument are expected to be larger or smaller, and the package then scales LTZ/UCI sensitivity along that pattern.

The package does not estimate an unrestricted direct-effect field. Instead, researchers supply a structured pattern and ask whether conclusions survive direct effects scaled along that pattern.

In applied work, users should usually vary `delta` over a range with `spliv_sensitivity_path()` rather than report one arbitrary sensitivity value.

Patterned sensitivity currently supports one endogenous treatment, one excluded instrument, and one researcher-specified pattern at a time.

Confirmatory BPE is not a subgroup-search procedure. The researcher must supply a pre-specified `bpe_design()` object, the package validates that subset, and BPE proceeds only if the confirmatory eligibility checks pass.

The first-stage F-statistic is still reported for diagnostics, but confirmatory BPE eligibility is determined by the pre-specification checks, subset size, cluster count, residualized instrument variation, and a first-stage equivalence interval.

BPE reduced-form covariance is propagated as a full covariance matrix and can optionally be inflated via `bpe_transport`.

Value

Object of class `plausexog_fit`.

Deprecated Wrappers

`plausexog()` and `plausexog_iv()` are deprecated compatibility aliases for `spliv()`.

Examples

```
set.seed(1)
d <- data.frame(y = rnorm(60), x = rnorm(60), z = rnorm(60), w = rnorm(60))
fit <- spliv(y ~ x + w | z + w, d, method = "uci", delta = 0.1)
fit$estimates
```

---

spliv_eval_pattern *Evaluate a Direct-Effect Pattern*

Description

Evaluates a `spliv_pattern()` object on a data frame and returns a numeric vector after optional centering and normalization.

Usage

```
spliv_eval_pattern(pattern, data)
```

Arguments

pattern	A spliv_pattern() object.
data	Data frame used for evaluation.

Value

Numeric vector of length nrow(data).

Examples

```
d <- data.frame(exposure = seq(-1, 1, length.out = 5))
p <- spliv_pattern("Exposure", ~ exposure,
  rationale = "Illustration of a monotone exposure pattern.")
spliv_eval_pattern(p, d)
```

---

spliv_pattern *Create a Patterned Exclusion-Violation Object*

Description

Creates a researcher-specified direct-effect pattern for patterned sensitivity analysis. The pattern determines where direct effects of the instrument are expected to be larger or smaller in the estimation sample.

Usage

```
spliv_pattern(
  name,
  pattern,
  rationale,
  variables_used = NULL,
  pattern_type = NULL,
  normalize = c("max_abs", "sd", "none"),
  center = FALSE,
  pre_specified = TRUE,
  notes = NULL
)
```

Arguments

name	Short pattern name.
pattern	Pattern specification. Accepts a one-sided formula, a function(data), a numeric vector of length nrow(data), a logical vector of length nrow(data), or a character string naming a numeric or logical column in data.
rationale	Substantive justification for the proposed pattern.
variables_used	Optional character vector describing the variables used to define the pattern.

pattern_type	Optional short label such as "theory_defined" or "design_based".
normalize	One of "max_abs" (default), "sd", or "none".
center	Logical; if TRUE, subtracts the estimation-sample mean before normalization.
pre_specified	Logical indicator for whether the pattern was chosen before examining outcome results.
notes	Optional free-form notes.

Value

An object of class "spliv_pattern".

Examples

```
p <- spliv_pattern(
  name = "Exposure", pattern = ~ exposure,
  rationale = "The alternative channel follows exposure.",
  variables_used = "exposure", pattern_type = "theory_defined"
)
p$name
```

---

spliv_sensitivity_path

Sensitivity Path over Delta Grid

Description

Runs spliv() repeatedly over a user-supplied delta_grid and returns a tidy sensitivity-path object. This is the recommended workflow for patterned or uniform sensitivity analysis: users should usually report how intervals change over a range of delta values rather than selecting one arbitrary sensitivity level.

Usage

```
spliv_sensitivity_path(
  formula,
  data,
  method = c("uci", "ltz"),
  delta_grid = seq(0, 0.2, by = 0.01),
  violation_pattern = NULL,
  stop_on_error = TRUE,
  ...
)
```

Arguments

formula	IV formula y ~ X | Z.
data	Data frame.
method	One of "uci" or "ltz". spliv_sensitivity_path() intentionally excludes confirmatory BPE.
delta_grid	Non-negative sensitivity grid. For UCI, each value d implies theta bounds [-d, +d].
violation_pattern	Optional spliv_pattern() object. If omitted, the path uses the package's backward-compatible uniform direct-effect pattern.
stop_on_error	Logical; if TRUE (default), stop on the first failed fit. If FALSE, record NA rows and store the error message in the returned error column.
...	Additional named arguments passed through to spliv(), such as fe, vcov, cluster, scale_instrument, or grid = list(level = 0.95).

Value

A data frame with class c("spliv_sensitivity_path", "data.frame"). The returned object includes path columns such as delta, method, estimate, conf_low, conf_high, contains_zero, and pattern metadata, plus attributes containing the original call, the supplied grid, and a tipping-point summary.

Examples

```
set.seed(5)
d <- data.frame(y = rnorm(80), x = rnorm(80), z = rnorm(80))
p <- spliv_sensitivity_path(y ~ x | z, d, method = "uci",
  delta_grid = c(0, 0.1), vcov = "hc1")
head(p)
```

---

spliv_tipping_point *Extract Tipping-Point Delta from a Sensitivity Path*

Description

Returns the tipping-point attribute stored on a spliv_sensitivity_path() object. For each term, the tipping point is the smallest delta on the supplied grid whose interval includes zero.

Usage

```
spliv_tipping_point(x)
```

Arguments

x	A spliv_sensitivity_path object.

Value

Named numeric vector of tipping-point values.

Examples

```
set.seed(6)
d <- data.frame(y = rnorm(80), x = rnorm(80), z = rnorm(80))
p <- spliv_sensitivity_path(y ~ x | z, d, method = "uci", delta_grid = c(0, 0.1))
spliv_tipping_point(p)
```

---

sp_ltz *Local-to-Zero Inference for Plausibly Exogenous IV*

Description

Formula-level wrapper over the matrix core LTZ implementation.

Usage

```
sp_ltz(
  formula,
  data,
  omega,
  mu,
  level = 0.95,
  vcov = c("hc1", "hc0", "iid", "cluster"),
  cluster = NULL
)

conley_ltz(
  formula,
  data,
  omega,
  mu,
  level = 0.95,
  vcov = c("hc1", "hc0", "iid", "cluster"),
  cluster = NULL
)
```

Arguments

formula	IV formula y ~ X | Z.
data	Data frame.
omega	Prior covariance matrix over instrument direct effects.
mu	Prior mean vector over instrument direct effects.
level	Confidence level.
vcov	One of "iid", "hc0", "hc1", "cluster".
cluster	Cluster ids when vcov = "cluster".

Value

Data frame with adjusted estimates and confidence intervals.

Examples

```
set.seed(3)
d <- data.frame(y = rnorm(80), x = rnorm(80), z = rnorm(80))
prior <- conley_prior_ltz(y ~ x | z, d, inst_vary = "z", sd = 0.1)
sp_ltz(y ~ x | z, d, prior$omega, prior$mu)
```

---

sp_prior_ltz *Build LTZ Prior Matrices for Chosen Instruments*

Description

Build LTZ Prior Matrices for Chosen Instruments

Usage

```
sp_prior_ltz(
  formula,
  data = NULL,
  inst_vary,
  mean = 0,
  sd = 1,
  vcov = c("hc1", "hc0", "iid", "cluster"),
  cluster = NULL
)

conley_prior_ltz(
  formula,
  data = NULL,
  inst_vary,
  mean = 0,
  sd = 1,
  vcov = c("hc1", "hc0", "iid", "cluster"),
  cluster = NULL
)
```

Arguments

formula	Formula or plausexog_fit.
data	Data frame when formula is a formula.
inst_vary	Instrument(s) with non-degenerate prior.
mean	Prior mean(s).
sd	Prior sd(s).
vcov	Vcov type.
cluster	Optional cluster ids.

Value

List with mu, omega, and instrument names.

Examples

```
set.seed(9)
d <- data.frame(y = rnorm(60), x = rnorm(60), z = rnorm(60))
sp_prior_ltz(y ~ x | z, d, inst_vary = "z", sd = 0.1)
```

---

sp_sensitivity_ltz_normal

LTZ Sensitivity over Delta Grid

Description

LTZ Sensitivity over Delta Grid

Usage

```
sp_sensitivity_ltz_normal(
  formula,
  data = NULL,
  term,
  inst_vary,
  delta_grid,
  mean_fun = function(delta) 0,
  sd_fun = function(delta) delta,
  level = 0.95,
  vcov = c("hc1", "hc0", "iid", "cluster"),
  cluster = NULL,
  scale_instrument = c("residual_sd", "none")
)

conley_sensitivity_ltz_normal(
  formula,
  data = NULL,
  term,
  inst_vary,
  delta_grid,
  mean_fun = function(delta) 0,
  sd_fun = function(delta) delta,
  level = 0.95,
  vcov = c("hc1", "hc0", "iid", "cluster"),
  cluster = NULL,
  scale_instrument = c("residual_sd", "none")
)
```

Arguments

formula	Formula or plausexog_fit.
data	Data frame when formula is a formula.
term	Coefficient name to track.
inst_vary	Instrument(s) with plausible violation.
delta_grid	Delta grid.
mean_fun	Function mapping delta to prior mean.
sd_fun	Function mapping delta to prior sd.
level	Confidence level.
vcov	Vcov type.
cluster	Optional cluster ids.
scale_instrument	One of "residual_sd" (default) or "none".

Value

Data frame with sensitivity path.

---

sp_sensitivity_ltz_uniform01_as_normal

LTZ Sensitivity with Normal Approximation to U(0, delta)

Description

LTZ Sensitivity with Normal Approximation to U(0, delta)

Usage

```
sp_sensitivity_ltz_uniform01_as_normal(
  formula,
  data = NULL,
  term,
  inst_vary,
  delta_grid,
  level = 0.95,
  vcov = c("hc1", "hc0", "iid", "cluster"),
  cluster = NULL,
  scale_instrument = c("residual_sd", "none")
)

conley_sensitivity_ltz_uniform01_as_normal(
  formula,
  data = NULL,
```

```
  term,
  inst_vary,
  delta_grid,
  level = 0.95,
  vcov = c("hc1", "hc0", "iid", "cluster"),
  cluster = NULL,
  scale_instrument = c("residual_sd", "none")
)
```

Arguments

formula	Formula or plausexog_fit.
data	Data frame when formula is a formula.
term	Coefficient name to track.
inst_vary	Instrument(s) with plausible violation.
delta_grid	Delta grid.
level	Confidence level.
vcov	Vcov type.
cluster	Optional cluster ids.
scale_instrument	One of "residual_sd" (default) or "none".

Value

Data frame with sensitivity path.

`sp_sensitivity_uci_support`
UCI Sensitivity over Delta Grid

Description

UCI Sensitivity over Delta Grid

Usage

```
sp_sensitivity_uci_support(
  formula,
  data = NULL,
  term,
  inst_vary,
  delta_grid,
  gmin_fun = function(delta) -delta,
  gmax_fun = function(delta) delta,
  grid = 41,
```

```
  level = 0.95,
  vcov = c("hc1", "hc0", "iid", "cluster"),
  cluster = NULL,
  scale_instrument = c("residual_sd", "none")
)

conley_sensitivity_uci_support(
  formula,
  data = NULL,
  term,
  inst_vary,
  delta_grid,
  gmin_fun = function(delta) -delta,
  gmax_fun = function(delta) delta,
  grid = 41,
  level = 0.95,
  vcov = c("hc1", "hc0", "iid", "cluster"),
  cluster = NULL,
  scale_instrument = c("residual_sd", "none")
)
```

Arguments

`formula`	Formula or `plausexog_fit`.
`data`	Data frame when `formula` is a formula.
`term`	Coefficient name to track.
`inst_vary`	Instrument(s) whose gamma bounds vary.
`delta_grid`	Delta grid.
`gmin_fun`	Function mapping delta to lower bound.
`gmax_fun`	Function mapping delta to upper bound.
`grid`	Points per instrument bound.
`level`	Confidence level.
`vcov`	Vcov type.
`cluster`	Optional cluster ids.
`scale_instrument`	One of `"residual_sd"` (default) or `"none"`.

Value

Data frame with sensitivity path.

sp_uci *Union of Confidence Intervals for Plausibly Exogenous IV*

Description

Formula-level wrapper over the matrix core UCI implementation.

Usage

```
sp_uci(
  formula,
  data,
  inst,
  gmin,
  gmax,
  grid = 21,
  level = 0.95,
  vcov = c("hc1", "hc0", "iid", "cluster"),
  cluster = NULL
)

conley_uci(
  formula,
  data,
  inst,
  gmin,
  gmax,
  grid = 21,
  level = 0.95,
  vcov = c("hc1", "hc0", "iid", "cluster"),
  cluster = NULL
)
```

Arguments

formula	IV formula `y ~ X | Z`.
data	Data frame.
inst	Instrument names to vary.
gmin	Lower bound(s) on gamma.
gmax	Upper bound(s) on gamma.
grid	Grid size per varied instrument.
level	Confidence level.
vcov	One of `"iid"`, `"hc0"`, `"hc1"`, `"cluster"`.
cluster	Cluster ids when `vcov = "cluster"`.

Value

Data frame with union confidence intervals.

Examples

```
set.seed(4)
d <- data.frame(y = rnorm(80), x = rnorm(80), z = rnorm(80))
sp_uci(y ~ x | z, d, inst = "z", gmin = -0.1, gmax = 0.1, grid = 5)
```

Index