

Package ‘confoundvis’

September 30, 2026

Type Package

Title Visualization Tools for Sensitivity Analysis of Unmeasured Confounding

Version 0.2.0

Description Visualization and reporting tools for sensitivity analysis to unmeasured confounding in observational studies. A common 'confoundsens' object stores a sensitivity path (the treatment effect as a function of hypothetical confounder strength) regardless of the framework that produced it, so the same robustness curves, contour plots, covariate benchmark ("sensitivity Love") plots, and plain-language reports can be drawn for impact threshold analysis (Frank, 2000, <doi:10.1177/0049124100029002001>), partial R-squared omitted-variable bias analysis (Cinelli and Hazlett, 2020, <doi:10.1111/rssb.12348>), and E-values (VanderWeele and Ding, 2017, <doi:10.7326/M16-2607>). Paths can be computed directly from fitted linear models or converted from results produced by the 'sensemakr', 'konfound', and 'EValue' packages.

License GPL-3

Encoding UTF-8

Language en-US

Depends R (>= 4.1.0)

Imports ggplot2 (>= 3.4.0), rlang, graphics, grid, stats

Suggests testthat (>= 3.0.0), knitr, rmarkdown, gridExtra, sensemakr, konfound, EValue

Config/testthat/edition 3

VignetteBuilder knitr

URL <https://github.com/subirhait/confoundvis>

BugReports <https://github.com/subirhait/confoundvis/issues>

NeedsCompilation no

Config/roxygen2/version 8.0.0

Author Subir Hait [aut, cre] (ORCID: <<https://orcid.org/0009-0004-9871-9677>>)
Maintainer Subir Hait <haitsubi@msu.edu>
Repository CRAN
Date/Publication 2026-09-30 21:21:02 UTC

Contents

as.data.frame.confoundsens	2
as_confoundsens	3
covariate_impacts	4
fit_local_quadratic	6
from_evalue	7
from_konfound	8
from_sensemakr	9
itcv_lm	10
new_confoundsens	11
plot_cone_comparison	12
plot_local_taylor	13
plot_reversal_cone	14
plot_robustness_curve	15
plot_sensitivity_contour	16
plot_sensitivity_love	17
plot_taylor_panels	18
print.confoundsens	19
robustness_points	19
sens_path_lm	20
sens_report	21
simulate_taylor_demo	22
summary.confoundsens	23
Index	24

as.data.frame.confoundsens
<i>Coerce a confoundsens object to a data frame</i>

Description

Coerce a confoundsens object to a data frame

Usage

```
## S3 method for class 'confoundsens'  
as.data.frame(x, ...)
```

Arguments

`x` A 'confoundsens' object.
`...` Unused; included for S3 method consistency.

Value

A data.frame with columns 'lambda' and 'theta', plus optional columns 'level', 'se', and 't' when present in 'x'.

Examples

```
x <- new_confoundsens(
  lambda = seq(0, 0.2, length.out = 5),
  theta  = seq(1, 0.8, length.out = 5),
  se      = rep(0.05, 5)
)
as.data.frame(x)
```

as_confoundsens	<i>Convert a data frame to a confoundsens object</i>
-----------------	--

Description

Converts a data frame containing sensitivity analysis results into a 'confoundsens' object suitable for use with 'confoundvis' plotting functions.

Usage

```
as_confoundsens(data, ...)

## Default S3 method:
as_confoundsens(data, ...)

## S3 method for class 'lm'
as_confoundsens(data, ...)

## S3 method for class 'sensemakr'
as_confoundsens(data, ...)

## S3 method for class 'data.frame'
as_confoundsens(
  data,
  lambda = "lambda",
  theta  = "theta",
  level  = NULL,
  se     = NULL,
  t      = NULL,
  ...
)
```

Arguments

<code>data</code>	A <code>data.frame</code> containing sensitivity analysis results, a fitted ‘lm’ model, or a ‘sensemakr’ object.
<code>...</code>	Passed to methods. For ‘lm’ objects, ‘treatment’ (required), ‘lambda’, and ‘ratio’; for ‘sensemakr’ objects, ‘lambda’ and ‘ratio’.
<code>lambda</code>	Character string; name of the column containing lambda values. Default “lambda”.
<code>theta</code>	Character string; name of the column containing theta values. Default “theta”.
<code>level</code>	Optional character string; name of the column containing level identifiers (e.g., “within” / “between”).
<code>se</code>	Optional character string; name of the column containing standard errors for ‘theta’.
<code>t</code>	Optional character string; name of the column containing test statistics.

Details

‘as_confoundsens()’ is generic. The data-frame method (the 0.1.0 behaviour) maps columns to fields. Methods for fitted ‘lm’ models and ‘sensemakr’ objects compute the partial R^2 path via [sens_path_lm()] and [from_sensemakr()]; see also [from_konfound()] and [from_evalue()].

Value

A ‘confoundsens’ object.

Examples

```
df <- data.frame(
  lambda = seq(0, 0.2, length.out = 10),
  theta = seq(1, 0.5, length.out = 10),
  se = rep(0.1, 10),
  level = rep(c("within", "between"), length.out = 10)
)
x <- as_confoundsens(df)
x

# Directly from a fitted model
fit <- lm(mpg ~ am + wt + hp, data = mtcars)
as_confoundsens(fit, treatment = "am")
```

covariate_impacts

Benchmark observed covariates against a sensitivity threshold

Description

For each covariate term in a fitted linear model (other than the treatment and the intercept), computes how strongly it is related to the treatment and to the outcome. These are the observed benchmarks that analysts compare with a sensitivity threshold ("would an omitted variable as strong as this covariate overturn the result?").

Usage

```
covariate_impacts(
  model,
  treatment,
  metric = c("itcv", "partial_r2"),
  alpha = 0.05
)
```

Arguments

model	A fitted ‘lm’ object.
treatment	Character string naming the treatment coefficient.
metric	Which metric populates the ‘impact’ column used by [plot_sensitivity_love()]: “itcv” (default) or “partial_r2”.
alpha	Two-sided significance level used for the ITCV threshold.

Details

Two metrics are returned:

‘impact’ (ITCV metric) $r_{DZ|W} \times r_{YZ|W}$, the product of partial correlations of the covariate with the treatment and with the outcome, each conditioning on the remaining covariates W (as in Frank, 2000). Compared with the ITCV. Only positive values (same sign as the estimate) push the estimate toward zero; the returned ‘impact’ is signed so that positive means “reduces the estimate”. Defined only for single-column terms.

‘bias_index’ (partial R^2 metric) $\sqrt{R_{Y \sim Z|D,W}^2 R_{D \sim Z|W}^2 / (1 - R_{D \sim Z|W}^2)}$. An omitted confounder with this pair of partial R^2 values would move the estimate by ‘bias_index’ $\times se\sqrt{df}$, so it would bring the point estimate to zero when ‘bias_index’ exceeds $|t|/\sqrt{df}$. Defined for multi-column terms (factors) as well.

The returned data frame carries an attribute ‘threshold’ for the selected ‘metric’, so it can be passed straight to [plot_sensitivity_love()]. Its ‘r_yu’, ‘r_du’, and ‘label’ columns can be passed as ‘benchmarks’ to [plot_sensitivity_contour()].

Covariate benchmarks describe *observed* variables. Whether an unobserved confounder could be as strong as a given covariate is a substantive judgment that no benchmark can settle.

Value

A data frame with one row per covariate term and columns ‘covariate’, ‘term_df’, ‘r_du’, ‘r_yu’, ‘itcv_impact’, ‘r2dz.x’, ‘r2yz.dx’, ‘bias_index’, ‘impact’, and ‘label’, and attributes ‘threshold’, ‘metric’, ‘itcv’, and ‘f_threshold’.

Examples

```
fit <- lm(mpg ~ am + wt + hp + qsec, data = mtcars)
imp <- covariate_impacts(fit, "am")
imp
plot_sensitivity_love(imp)
```

fit_local_quadratic	<i>Fit a local quadratic approximation</i>
---------------------	--

Description

Fits a second-order Taylor (quadratic) approximation of an effect path $\theta(\delta)$ near $\delta = 0$ using ordinary least squares:

$$\theta(\delta) \approx a + b\delta + c\delta^2.$$

Usage

```
fit_local_quadratic(
  data = NULL,
  delta = NULL,
  theta = NULL,
  local_max_delta = 0.2,
  include_intercept = TRUE,
  tol = 1e-12
)
```

Arguments

data	Optional data.frame containing columns named ‘delta’ and ‘theta’. If supplied, the ‘delta’ and ‘theta’ arguments are ignored.
delta	Optional numeric vector of delta values. Used only when ‘data = NULL’.
theta	Optional numeric vector of theta values. Used only when ‘data = NULL’.
local_max_delta	Positive numeric scalar giving the half-width of the local window: only observations with $ \delta \leq \text{‘local_max_delta’}$ are used in the fit.
include_intercept	Logical; if ‘TRUE’ (default), include an intercept term.
tol	Non-negative numeric scalar tolerance used when selecting observations within the local window.

Details

You may supply either a data.frame containing columns ‘delta’ and ‘theta’, or supply numeric vectors ‘delta’ and ‘theta’ directly.

Value

A named list with elements: * ‘coef’ — named coefficient vector from [stats::lm()]. * ‘intercept’, ‘slope’, ‘quad’ — individual coefficients (NA when absent). * ‘model’ — the fitted ‘lm’ object. * ‘local_data’ — the data.frame used for fitting. * ‘local_max_delta’ — the window half-width used.

Examples

```
df <- data.frame(
  delta = seq(0, 0.3, length.out = 60),
  theta = 0.4 - 0.7 * seq(0, 0.3, length.out = 60) +
    0.4 * seq(0, 0.3, length.out = 60)^2
)
fit_local_quadratic(df, local_max_delta = 0.2)
```

from_evalue

Convert an E-value result to a confoundsens path

Description

Builds a bias-adjusted risk-ratio path from an ‘evalue’ object (as returned by ‘EValue::evaluates.RR()’, ‘evaluates.OR()’, ‘evaluates.HR()’, ‘evaluates.OLS()’, and related functions). Confounder strength λ is the common risk ratio relating the confounder to both treatment and outcome; the corresponding bias factor is $B = \lambda^2 / (2\lambda - 1)$ (VanderWeele and Ding, 2017). The path is on the log risk-ratio scale, so it crosses the null (0) exactly at the E-value.

Usage

```
from_evalue(x, lambda = seq(1, 5, by = 0.02))
```

Arguments

x	An object of class “evalue”.
lambda	Numeric vector of confounder risk ratios (≥ 1).

Value

A ‘confoundsens’ object with ‘theta’ = log of the bias-adjusted point risk ratio and, when the confidence interval is available, ‘se’ on the log scale.

References

VanderWeele, T. J., & Ding, P. (2017). Sensitivity analysis in observational research: Introducing the E-value. **Annals of Internal Medicine**, 167(4), 268–274. doi:10.7326/M162607

Examples

```
e <- EValue::evaluates.RR(est = 1.8, lo = 1.3, hi = 2.5)
path <- from_evalue(e)
robustness_points(path)
```

from_konfound	<i>Convert konfound output to a confoundsens path</i>
---------------	---

Description

Builds the ITCV sensitivity path from the observed and critical correlations reported by ‘konfound::konfound()’ or ‘konfound::pkonfound()’ when called with ‘to_return = "raw_output"’.

Usage

```
from_konfound(
  x,
  dof = NULL,
  lambda = seq(0, 0.5, by = 0.01),
  treatment = NA_character_
)
```

Arguments

x	A list returned by ‘konfound()’/‘pkonfound()’ with ‘to_return = "raw_output"’; must contain ‘obs_r’ and ‘critical_r’.
dof	Optional residual degrees of freedom. If supplied, the path also carries adjusted t statistics.
lambda	Numeric vector of impacts in $[0, 1)$.
treatment	Optional label for the treatment.

Value

A ‘confoundsens’ object on the correlation scale; ‘meta\$itcv’ holds the conditional ITCV implied by ‘obs_r’ and ‘critical_r’ (konfound’s ‘itcvGz’).

See Also

[itcv_lm()]

Examples

```
k <- konfound::pkonfound(est_eff = 2, std_err = 0.4, n_obs = 100,
  n_covariates = 3, index = "IT",
  to_return = "raw_output")
path <- from_konfound(k)
path$meta$itcv
```

from_sensemakr	<i>Convert a sensemakr result to a confoundsens path</i>
----------------	--

Description

Builds the partial R^2 sensitivity path from the summary statistics stored in a ‘sensemakr’ object (estimate, standard error, and residual degrees of freedom), using the same bias formulas as [sens_path_lm()]. Benchmark bounds stored in the object are attached as ‘meta\$bounds’.

Usage

```
from_sensemakr(x, lambda = seq(0, 0.5, by = 0.01), ratio = 1)
```

Arguments

x	An object of class “sensemakr” (from ‘sensemakr::sensemakr()’).
lambda	Numeric vector of confounder strengths in $[0, 1)$. Default is a grid from 0 to 0.5.
ratio	Positive scalar; ratio of the outcome-side to the treatment-side partial R^2 .

Value

A ‘confoundsens’ object.

See Also

[sens_path_lm()]

Examples

```
data("darfur", package = "sensemakr")
fit <- lm(peacefactor ~ directlyharmed + age + farmer_dar + herder_dar +
  pastvoted + hhsize_darfur + female + village, data = darfur)
s <- sensemakr::sensemakr(fit, treatment = "directlyharmed",
  benchmark_covariates = "female", kd = 1:3)
path <- from_sensemakr(s)
sens_report(path)
```

itcv_lm

*Impact threshold for a confounding variable from a fitted linear model***Description**

Computes Frank's (2000) impact threshold for a confounding variable (ITCV) for one coefficient of a linear model, together with the ITCV sensitivity path: the partial correlation between treatment and outcome after conditioning on a confounder whose impact $k = r_{DU}r_{YU}$ is split equally between the two correlations ($r_{DU} = r_{YU} = \sqrt{k}$).

Usage

```
itcv_lm(model, treatment, alpha = 0.05, lambda = seq(0, 0.5, by = 0.01))
```

Arguments

model	A fitted 'lm' object.
treatment	Character string naming the treatment coefficient.
alpha	Two-sided significance level used for the critical value.
lambda	Numeric vector of impacts in $[0, 1)$ at which to evaluate the path.

Details

Along this path the adjusted correlation is $(r - k)/(1 - k)$, which equals the critical correlation $r^\#$ exactly at $k = \text{ITCV} = (|r| - |r^\#|)/(1 - |r^\#|)$.

The conditional ITCV equals 'itcvGz' and the unconditional ITCV equals 'itcv' in the raw output of 'konfound::konfound()' (version 1.0).

Value

A list with elements 'r' (observed partial correlation), 'r_crit' (critical partial correlation), 'itcv' (the ITCV for a confounder's correlations *conditional on* the model covariates), 'dof', 'path' (a 'confoundsens' object on the correlation scale), 'r2xz' and 'r2yz' (the R^2 of the covariates with treatment and outcome), and 'itcv_unconditional' (the ITCV expressed in unconditional correlations, $\text{itcv} * \sqrt{(1 - r2xz) * (1 - r2yz)}$).

References

Frank, K. A. (2000). Impact of a confounding variable on a regression coefficient. *Sociological Methods & Research*, 29(2), 147–194. doi:10.1177/0049124100029002001

Frank, K. A., Maroulis, S. J., Duong, M. Q., & Kelcey, B. M. (2013). What would it take to change an inference? Using Rubin's causal model to interpret the robustness of causal inferences. *Educational Evaluation and Policy Analysis*, 35(4), 437–460. doi:10.3102/0162373713493129

See Also

```
[from_konfound()], [sens_path_lm()]
```

Examples

```
fit <- lm(mpg ~ am + wt + hp, data = mtcars)
it <- itcv_lm(fit, "am")
it$itcv
plot_robustness_curve(it$path, points = FALSE)
```

new_confoundsens	<i>Create a confoundsens object</i>
------------------	-------------------------------------

Description

Constructs a ‘confoundsens’ object — a lightweight container for storing sensitivity paths (e.g., $\theta(\lambda)$) and optional uncertainty or stratification information used by ‘confoundvis’ plotting functions.

Usage

```
new_confoundsens(
  lambda,
  theta,
  level = NULL,
  se = NULL,
  t = NULL,
  meta = list()
)
```

Arguments

lambda	Numeric vector of sensitivity strength values (e.g., ITCV lambda or delta). Should be nondecreasing; a warning is issued otherwise.
theta	Numeric vector of effect estimates along the sensitivity path. Must be the same length as ‘lambda’.
level	Optional character (or coercible) vector of level identifiers (e.g., “within”, “between”). Must be the same length as ‘lambda’.
se	Optional numeric vector of standard errors for ‘theta’. Must be the same length as ‘lambda’.
t	Optional numeric vector of test statistics along the path. Must be the same length as ‘lambda’.
meta	Optional named list of metadata describing the path, for example ‘framework’ (“partial_r2”, “itcv”, “evaluate”, ...), ‘lambda_label’, ‘theta_label’, ‘null’ (the value of ‘theta’ that represents no effect), and ‘estimate’. Used by printing, plotting, and [sens_report()]; ignored otherwise.

Value

A ‘confoundsens’ object (a list with class “confoundsens”).

Examples

```
x <- new_confoundsens(
  lambda = seq(0, 0.2, length.out = 10),
  theta  = seq(1, 0.6, length.out = 10),
  se      = rep(0.1, 10)
)
x
```

plot_cone_comparison *Compare reversal cones across multilevel components*

Description

Produces a two-panel plot comparing the reversal cone cross-sections for the within-cluster and between-cluster confounding components in multilevel settings. Each panel calls [plot_reversal_cone()] and they are displayed side by side using faceting.

Usage

```
plot_cone_comparison(
  theta0_within = 0.35,
  theta0_between = 0.5,
  delta_within  = 0.2,
  delta_between = 0.3,
  ...
)
```

Arguments

theta0_within Numeric; within-cluster baseline effect.
 theta0_between Numeric; between-cluster baseline effect.
 delta_within Numeric; within-cluster confounding magnitude (> 0).
 delta_between Numeric; between-cluster confounding magnitude (> 0).
 ... Additional arguments passed to [plot_reversal_cone()].

Value

Invisibly, a list with ggplot objects ‘within’ and ‘between’. The plots are drawn side-by-side to the current device.

Examples

```
plot_cone_comparison(
  theta0_within = 0.35, theta0_between = 0.50,
  delta_within  = 0.20, delta_between  = 0.30
)
```

plot_local_taylor	<i>Local Taylor diagnostic plot</i>
-------------------	-------------------------------------

Description

Plots local Taylor series components (or any multi-series decomposition) as a function of ‘delta’ from a long-form data.frame. Lines are distinguished by colour and linetype, keyed by ‘series’.

Usage

```
plot_local_taylor(df, facet = FALSE)
```

Arguments

df	A data.frame in long form with columns: * ‘delta’ — numeric; the confounding-strength grid. * ‘series’ — character or factor; name of each curve (e.g., “path”, “tangent”, “quadratic”). * ‘value’ — numeric; the effect value for each (delta, series) pair.
facet	Logical; if ‘TRUE’, produce a faceted plot with one panel per ‘series’. If ‘FALSE’ (default), overlay all series on a single panel with colour and linetype aesthetics.

Value

A [ggplot2::ggplot()] object.

Examples

```
df <- data.frame(
  delta = rep(seq(0, 0.2, length.out = 25), 3),
  series = rep(c("path", "tangent", "quadratic"), each = 25),
  value = c(
    0.4 - 0.7 * seq(0, 0.2, length.out = 25) - 0.4 * seq(0, 0.2, length.out = 25)^2,
    0.4 - 0.7 * seq(0, 0.2, length.out = 25),
    0.4 - 0.7 * seq(0, 0.2, length.out = 25) + 0.2 * seq(0, 0.2, length.out = 25)^2
  )
)
plot_local_taylor(df)
plot_local_taylor(df, facet = TRUE)
```

plot_reversal_cone *Reversal cone geometry plot*

Description

Visualizes the two-dimensional cross-section of the reversal cone C_l at a fixed confounding effect magnitude $|\delta|$. The cone partitions the (q, p) confounding parameter space into an **attenuation zone** (where the effect shrinks but does not reverse sign) and a **reversal zone** (where the sign changes). The boundary between zones is the reversal curve.

Usage

```
plot_reversal_cone(
  theta0 = 0.4,
  delta = 0.2,
  q_range = c(0, 1),
  p_range = c(-1, 1),
  grid_n = 200L,
  show_boundary = TRUE,
  show_volume = TRUE
)
```

Arguments

theta0	Numeric; baseline estimated effect at zero confounding (must be nonzero).
delta	Numeric; the fixed confounding effect magnitude $ \delta $ at which the cross-section is evaluated (> 0).
q_range	Numeric vector of length 2; range of the confounding prevalence parameter $q \in [0, 1]$ (default 'c(0, 1)').
p_range	Numeric vector of length 2; range of the confounding impact parameter p (default 'c(-1, 1)').
grid_n	Integer; grid resolution for each axis (default 200).
show_boundary	Logical; draw the reversal boundary curve.
show_volume	Logical; annotate with the share of the displayed grid lying in the reversal zone.

Details

The display uses the stylized bilinear bias model $\theta_{adj} = \theta_0 - |\delta| q p$, so the reversal boundary is the hyperbola $p = \theta_0 / (|\delta| q)$. This is a conceptual map of the parameter space, not an estimate: the annotated "reversal share" is the fraction of the **displayed** grid that falls in the reversal zone and therefore depends on 'q_range' and 'p_range'. It should not be reported as a probability or as a property of the data.

Value

A ggplot object.

Examples

```
plot_reversal_cone(theta0 = 0.40, delta = 0.20)
plot_reversal_cone(theta0 = 0.40, delta = 0.50)
```

plot_robustness_curve *Plot a robustness curve*

Description

Plots the sensitivity path stored in a ‘confoundsens’ object as a function of ‘lambda’. By default, plots the effect path ‘theta(lambda)’; optionally plots the test-statistic path ‘t(lambda)’. Pointwise 95 are drawn when standard errors are available.

Usage

```
plot_robustness_curve(
  x,
  what = c("theta", "t"),
  bands = TRUE,
  points = TRUE,
  facet_level = TRUE,
  reference = TRUE
)
```

Arguments

x	A ‘confoundsens’ object created by [new_confoundsens()] or [as_confoundsens()].
what	Character; which path to plot: “theta” (default) or “t”.
bands	Logical; if ‘TRUE’ and ‘x\$se’ is present, draw 95 confidence bands (applies only when ‘what = “theta”’).
points	Logical; if ‘TRUE’, overlay points on the line.
facet_level	Logical; if ‘TRUE’ and ‘x\$level’ is present, facet the plot by level.
reference	Logical; if ‘TRUE’ (default) and ‘x\$meta’ records a null value (‘meta\$null’) or a critical value (‘meta\$threshold’), draw them as horizontal reference lines (dotted and dashed respectively). Objects created by [new_confoundsens()] without ‘meta’ are unaffected.

Value

A [ggplot2::ggplot()] object.

Examples

```
x <- new_confoundsens(
  lambda = seq(0, 0.2, length.out = 25),
  theta = 1 - 2 * seq(0, 0.2, length.out = 25),
  se = rep(0.05, 25),
  level = rep(c("within", "between"), length.out = 25)
)
plot_robustness_curve(x)
plot_robustness_curve(x, bands = FALSE, facet_level = FALSE)
```

plot_sensitivity_contour

Sensitivity contour plot

Description

Draws an ITCV-style hyperbolic boundary in (r_{YU}, r_{DU}) space and optionally overlays observed covariate benchmarks as labelled points. The robust region is the interior of the hyperbola where $|r_{YU} \cdot r_{DU}| < \text{'threshold'}$.

Usage

```
plot_sensitivity_contour(threshold, grid_n = 200, benchmarks = NULL)
```

Arguments

threshold	Positive numeric scalar; the ITCV-style product threshold. The boundary satisfies $ r_{YU} \cdot r_{DU} = \text{'threshold'}$.
grid_n	Integer ≥ 50 ; number of points used per branch of the boundary curve. Larger values give smoother curves.
benchmarks	Optional data.frame with columns 'r_yu' and 'r_du' (numeric) and an optional 'label' column (character). Each row is plotted as a labelled benchmark point.

Value

A [ggplot2::ggplot()] object.

Examples

```
b <- data.frame(
  r_yu = c(0.10, 0.15),
  r_du = c(0.20, 0.12),
  label = c("SES", "BMI")
)
plot_sensitivity_contour(threshold = 0.02, benchmarks = b)
```

plot_sensitivity_love *Sensitivity Love plot*

Description

Benchmarks a sensitivity threshold against the empirical distribution of observed covariate impacts in a "Love plot"-style display. Each covariate appears as a point on a horizontal impact axis; the sensitivity threshold is shown as a vertical reference line. Covariates to the left of the line are weaker than the threshold; those to the right pose a credible threat.

Usage

```
plot_sensitivity_love(
  df,
  threshold = attr(df, "threshold"),
  sort = TRUE,
  top = NULL
)
```

Arguments

df	A data.frame with at least two columns: * 'covariate' — covariate names (character or factor). * 'impact' — numeric impact values (e.g., ITCV product $ r_{YU} \cdot r_{DU} $, partial R^2 , or other confounding-strength metric).
threshold	Single non-missing numeric value. Defaults to the "threshold" attribute of 'df', which [covariate_impacts()] sets. Drawn as a vertical reference line (the sensitivity threshold). Rows with missing 'impact' are dropped.
sort	Logical; if 'TRUE' (default), sort covariates by 'impact' on the y-axis (ascending).
top	Optional positive integer. If supplied, only the 'top' covariates with the largest absolute impact are displayed.

Value

A [ggplot2::ggplot()] object.

Examples

```
df <- data.frame(
  covariate = c("SES", "BMI", "Gender", "Race", "Age"),
  impact    = c(0.12, 0.05, 0.02, 0.08, 0.03)
)
plot_sensitivity_love(df, threshold = 0.10)
plot_sensitivity_love(df, threshold = 0.10, top = 3)

# From a fitted model: threshold is taken from covariate_impacts()
fit <- lm(mpg ~ am + wt + hp + qsec, data = mtcars)
plot_sensitivity_love(covariate_impacts(fit, "am"))
```

plot_taylor_panels *Three-panel Taylor approximation illustration*

Description

Produces a three-panel figure (linear / concave-down / convex-up) showing a simulated confounding path together with its first-order (tangent) and second-order (local quadratic) approximations, for three curvature regimes: linear, concave-down, and convex-up.

Usage

```
plot_taylor_panels(
  delta_max = 1.5,
  step = 0.02,
  theta0 = 0.4,
  slope = -0.7,
  kappa = 0.4,
  local_max_delta = 0.2
)
```

Arguments

delta_max	Positive numeric scalar. Upper bound of the delta grid.
step	Positive numeric scalar. Grid step size.
theta0	Numeric scalar. Baseline effect at delta = 0.
slope	Numeric scalar. First-order slope at delta = 0.
kappa	Non-negative numeric scalar. Curvature magnitude.
local_max_delta	Positive numeric scalar <= 'delta_max'. Width of the local window used for the quadratic approximation.

Details

This is a didactic illustration built from simulated paths ([simulate_taylor_demo()]); it does not analyse data. It was called 'plot_figure2_taylor_panels()' in version 0.1.0; that name still works but is deprecated.

Value

A named list with ggplot objects 'A' (linear), 'B' (concave), and 'C' (convex), returned invisibly. The three panels are also printed to the active graphics device via [gridExtra::grid.arrange()] if **gridExtra** is installed, or via [graphics::layout()] otherwise.

Examples

```
plots <- plot_taylor_panels()
# Access individual panels
plots$A
```

print.confoundsens	<i>Print method for confoundsens objects</i>
--------------------	--

Description

Prints a concise summary of the key fields in a ‘confoundsens’ object.

Usage

```
## S3 method for class 'confoundsens'
print(x, ...)
```

Arguments

x	A ‘confoundsens’ object.
...	Unused; included for S3 method consistency.

Value

‘x’, invisibly.

Examples

```
x <- new_confoundsens(
  lambda = seq(0, 0.2, length.out = 5),
  theta = seq(1, 0.8, length.out = 5)
)
print(x)
```

robustness_points	<i>Locate robustness thresholds along a sensitivity path</i>
-------------------	--

Description

Finds, for each level of a ‘confoundsens’ object, the confounder strength λ at which (i) the point estimate reaches the null value and (ii) statistical significance is lost. Crossings are located by linear interpolation between grid points, so their precision depends on the spacing of ‘lambda’.

Usage

```
robustness_points(x, alpha = 0.05)
```

Arguments

x	A ‘confoundsens’ object.
alpha	Two-sided significance level.

Details

The significance criterion is chosen from what the object carries, in this order: a critical value on the ‘theta’ scale in ‘meta\$threshold’ (ITCV paths); adjusted t statistics with residual degrees of freedom in ‘meta\$dof’ (partial R^2 paths; critical value from a t distribution with ‘dof - 1’ degrees of freedom, as in Cinelli and Hazlett, 2020); standard errors (normal critical value); or t statistics without degrees of freedom (normal critical value).

Value

A data frame with columns ‘level’, ‘theta_start’, ‘lambda_null’, ‘lambda_sig’, and ‘sig_rule’. ‘NA’ means the crossing is not reached within the supplied ‘lambda’ range.

Examples

```
fit <- lm(mpg ~ am + wt + hp, data = mtcars)
robustness_points(sens_path_lm(fit, "am", lambda = seq(0, 0.9, by = 0.001)))
```

sens_path_lm	<i>Sensitivity path from a fitted linear model (partial R-squared framework)</i>
--------------	--

Description

Computes the bias-adjusted treatment coefficient as a function of the strength of a hypothetical omitted confounder, using the omitted-variable bias formulas of Cinelli and Hazlett (2020). Confounder strength is indexed by λ , the partial R^2 of the confounder with the treatment ($R^2_{D \sim Z|X}$); the partial R^2 with the outcome is $R^2_{Y \sim Z|D,X} = \min(\text{ratio} \times \lambda, 1)$. The default ‘ratio = 1’ gives the equal-strength path along which the robustness value is defined.

Usage

```
sens_path_lm(model, treatment, lambda = seq(0, 0.5, by = 0.01), ratio = 1)
```

Arguments

model	A fitted ‘lm’ object.
treatment	Character string naming the treatment coefficient (a column of the model matrix).
lambda	Numeric vector of confounder strengths in $\backslash[0, 1)$. Default is a grid from 0 to 0.5.
ratio	Positive scalar; ratio of the outcome-side to the treatment-side partial R^2 .

Details

The confounder is assumed to act in the direction that shrinks the estimate toward zero (the conventional worst case for a reported effect).

Value

A ‘confoundsens’ object with ‘theta’ (adjusted estimate), ‘se’ (adjusted standard error), and ‘t’ (adjusted t statistic), plus ‘meta’ recording the original estimate, degrees of freedom, and robustness value.

References

Cinelli, C., & Hazlett, C. (2020). Making sense of sensitivity: Extending omitted variable bias. *Journal of the Royal Statistical Society: Series B*, 82(1), 39–67. doi:[10.1111/rssb.12348](https://doi.org/10.1111/rssb.12348)

See Also

[from_sensemakr()], [itcv_lm()], [covariate_impacts()]

Examples

```
fit <- lm(mpg ~ am + wt + hp, data = mtcars)
path <- sens_path_lm(fit, treatment = "am")
path
plot_robustness_curve(path, points = FALSE)
```

sens_report

Plain-language sensitivity report

Description

Summarises a ‘confoundsens’ path in a few sentences: the starting estimate, the framework-specific summary (robustness value, ITCV, or E-value) when available, and the confounder strengths at which the point estimate reaches the null and significance is lost. The report closes with a statement of what the analysis cannot establish.

Usage

```
sens_report(x, alpha = 0.05, digits = 3)
```

Arguments

x	A ‘confoundsens’ object.
alpha	Two-sided significance level.
digits	Number of significant digits.

Value

An object of class ‘confoundvis_report’: a list with ‘text’ (a character vector) and ‘points’ (the output of [robustness_points()]). It prints as text.

Examples

```
fit <- lm(mpg ~ am + wt + hp, data = mtcars)
sens_report(sens_path_lm(fit, "am", lambda = seq(0, 0.9, by = 0.001)))
```

simulate_taylor_demo *Simulate demo confounding paths for Taylor panel figures*

Description

Generates three synthetic sensitivity paths — linear, concave-down, and convex-up — sharing the same baseline effect $\theta(0)$ and first-order slope, but differing in second-order curvature. These toy paths illustrate the difference between tangent-based (first-order) and local quadratic (second-order) sensitivity approximations.

Usage

```
simulate_taylor_demo(
  delta_max = 1.5,
  step = 0.02,
  theta0 = 0.4,
  slope = -0.7,
  kappa = 0.4
)
```

Arguments

delta_max	Single positive numeric scalar. Upper bound of the δ grid.
step	Single positive numeric scalar. Step size for the grid.
theta0	Single numeric scalar. Baseline effect at $\delta = 0$.
slope	Single numeric scalar. First-order slope at $\delta = 0$.
kappa	Single non-negative numeric scalar. Curvature magnitude.

Details

The three regimes are:

- **Linear:** $\theta(\delta) = \theta_0 + s\delta$
- **Concave-down:** $\theta(\delta) = \theta_0 + s\delta - \kappa\delta^2$
- **Convex-up:** $\theta(\delta) = \theta_0 + s\delta + \kappa\delta^2$

Value

A named list with elements 'linear', 'concave', and 'convex', each a [new_confoundsens()] object.

Examples

```
sims <- simulate_taylor_demo(
  delta_max = 1, step = 0.05, theta0 = 0.4, slope = -0.7, kappa = 0.4
)
sims$linear
plot_robustness_curve(sims$concave)
```

summary.confoundsens	<i>Summary method for confoundsens objects</i>
----------------------	--

Description

Produces a concise numerical summary of the sensitivity path, optionally broken down by level.

Usage

```
## S3 method for class 'confoundsens'
summary(object, ...)
```

Arguments

object	A 'confoundsens' object.
...	Unused; included for S3 method consistency.

Value

An object of class 'summary.confoundsens' containing a 'table' data.frame with per-level summary statistics.

Examples

```
x <- new_confoundsens(
  lambda = seq(0, 0.2, length.out = 10),
  theta = seq(1, 0.6, length.out = 10),
  se = rep(0.08, 10),
  level = rep(c("within", "between"), length.out = 10)
)
summary(x)
```

Index

`as.data.frame.confoundsens`, [2](#)
`as.confoundsens`, [3](#)

`covariate_impacts`, [4](#)

`fit_local_quadratic`, [6](#)
`from_evalue`, [7](#)
`from_konfound`, [8](#)
`from_sensemakr`, [9](#)

`itcv_lm`, [10](#)

`new.confoundsens`, [11](#)

`plot_cone_comparison`, [12](#)
`plot_figure2_taylor_panels`
 (`plot_taylor_panels`), [18](#)
`plot_local_taylor`, [13](#)
`plot_reversal_cone`, [14](#)
`plot_robustness_curve`, [15](#)
`plot_sensitivity_contour`, [16](#)
`plot_sensitivity_love`, [17](#)
`plot_taylor_panels`, [18](#)
`print.confoundsens`, [19](#)

`robustness_points`, [19](#)

`sens_path_lm`, [20](#)
`sens_report`, [21](#)
`simulate_taylor_demo`, [22](#)
`summary.confoundsens`, [23](#)