Package ‘konfound’

May 28, 2025

Type Package

Title Quantify the Robustness of Causal Inferences

Version 1.0.3

Description Statistical methods that quantify the conditions necessary to alter inferences, also known as sensitivity analysis, are becoming increasingly important to a variety of quantitative sciences. A series of recent works, including Frank (2000) <doi:10.1177/0049124100029002001> and Frank et al. (2013) <doi:10.3102/0162373713493129> extend previous sensitivity analyses by considering the characteristics of omitted variables or unobserved cases that would change an inference if such variables or cases were observed. These analyses generate statements such as ``an omitted variable would have to be correlated at xx with the predictor of interest (e.g., the treatment) and outcome to invalidate an inference of a treatment effect". Or ``one would have to replace pp percent of the observed data with nor which the treatment had no effect to invalidate the inference".
We implement these recent developments of sensitivity analysis and provide modules to calculate these two robustness indices and generate such statements in R. In particular, the functions konfound(), pkonfound() and mkonfound() allow users to calculate the robustness of inferences for a user's own model, a single published study and multiple studies respectively.

URL `https://github.com/konfound-project/konfound`, `https://konfound-it.org/konfound/`

BugReports `https://github.com/konfound-project/konfound/issues`

Depends R (>= 3.5.0)

Imports broom, broom.mixed, crayon, dplyr, ggplot2, lavaan, purrr, rlang, tidyr, lme4 (>= 1.1-35.1), tibble, ggrepel, pbkrtest, ppcor

Suggests covr, devtools, forcats, knitr, rmarkdown, mice, roxygen2, testthat, Matrix (>= 1.6-2)

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Joshua M Rosenberg [aut, cre],
Ran Xu [ctb],
Qinyun Lin [ctb],
Spiro Maroulis [ctb],
Sarah Narvaiz [ctb],
Kenneth A Frank [ctb],
Wei Wang [ctb],
Yunhe Cui [ctb],
Gaofei Zhang [ctb],
Xuesen Cheng [ctb],
JiHoon Choi [ctb],
Guan Saw [ctb]

Maintainer Joshua M Rosenberg <`jmrosen48@gmail.com`>

Repository CRAN

Date/Publication 2025-05-28 20:30:09 UTC

Contents

`binary_dummy_data` *Binary dummy data*

Description

This data is made-up data for use in examples.

Format

A data.frame with 107 rows and 3 variables.

`cal_delta_star` *Calculate delta star for sensitivity analysis*

Description

Calculate delta star for sensitivity analysis

Usage

```
cal_delta_star(
  FR2max,
  R2,
  R2_uncond,
  est_eff,
  eff_thr,
  var_x,
  var_y,
  est_uncond,
  rxz,
  n_obs
)
```

Arguments

`FR2max`	maximum R2
`R2`	current R2
`R2_uncond`	unconditional R2
`est_eff`	estimated effect
`eff_thr`	effect threshold
`var_x`	variance of X
`var_y`	variance of Y
`est_uncond`	unconditional estimate
`rxz`	correlation coefficient between X and Z
`n_obs`	number of observations

Value

delta star value

`cal_rxy` *Calculate rxy based on ryxGz, rxz, and ryz*

Description

Calculate rxy based on ryxGz, rxz, and ryz

Usage

```
cal_rxy(ryxGz, rxz, ryz)
```

Arguments

`ryxGz`	correlation coefficient between Y and X given Z
`rxz`	correlation coefficient between X and Z
`ryz`	correlation coefficient between Y and Z

Value

rxy value

`cal_rxz` *Calculate R2xz based on variances and standard error*

Description

Calculate R2xz based on variances and standard error

Usage

```
cal_rxz(var_x, var_y, R2, df, std_err)
```

Arguments

`var_x`	variance of X
`var_y`	variance of Y
`R2`	coefficient of determination
`df`	degrees of freedom
`std_err`	standard error

Value

R2xz value

`cal_ryz` *Calculate R2yz based on ryxGz and R2*

Description

Calculate R2yz based on ryxGz and R2

Usage

```
cal_ryz(ryxGz, R2)
```

Arguments

`ryxGz`	correlation coefficient between Y and X given Z
`R2`	coefficient of determination

Value

R2yz value

`chisq_p` *Perform a Chi-Square Test*

Description

‘chisq_p‘ calculates the p-value for a chi-square test given a contingency table.

Usage

```
chisq_p(a, b, c, d)
```

Arguments

`a`	Frequency count for row 1, column 1.
`b`	Frequency count for row 1, column 2.
`c`	Frequency count for row 2, column 1.
`d`	Frequency count for row 2, column 2.

Value

P-value from the chi-square test.

`concord1` *Concord1 data*

Description

This data is from Hamilton (1983)

Format

A data.frame with 496 rows and 10 variables.

References

Hamilton, Lawrence C. 1983. Saving water: A causal model of household conservation. Sociological Perspectives 26(4):355-374.

`get_kr_df` *Extract Degrees of Freedom for Fixed Effects in a Linear Mixed-Effects Model*

Description

Extract Degrees of Freedom for Fixed Effects in a Linear Mixed-Effects Model

Usage

```
get_kr_df(model_object)
```

Arguments

`model_object`	The mixed-effects model object produced by lme4::lmer.

Value

A vector containing degrees of freedom for the fixed effects in the model.

`konfound` *Konfound Analysis for Various Model Types*

Description

Performs sensitivity analysis on fitted models including linear models (‘lm‘), generalized linear models (‘glm‘), and linear mixed-effects models (‘lmerMod‘). It calculates the amount of bias required to invalidate or sustain an inference,and the impact of an omitted variable necessary to affect the inference.

Usage

```
konfound(
  model_object,
  tested_variable,
  alpha = 0.05,
  tails = 2,
  index = "RIR",
  to_return = "print",
  two_by_two = FALSE,
  n_treat = NULL,
  switch_trm = TRUE,
  replace = "control"
)
```

Arguments

`model_object`	A model object produced by ‘lm‘, ‘glm‘, or ‘lme4::lmer‘.
`tested_variable`	Variable associated with the coefficient to be tested.
`alpha`	Significance level for hypothesis testing.
`tails`	Number of tails for the test (1 or 2).
`index`	Type of sensitivity analysis (’RIR’ by default).
`to_return`	Type of output to return (’print’, ’raw_output’, ’table’).
`two_by_two`	Boolean; if ‘TRUE‘, uses a 2x2 table approach for ‘glm‘ dichotomous variables.
`n_treat`	Number of treatment cases (used only if ‘two_by_two‘ is ‘TRUE‘).
`switch_trm`	Boolean; switch treatment and control in the analysis.
`replace`	Replacement method for treatment cases (’control’ by default).

Value

Depending on ‘to_return‘, prints the result, returns a raw output, or a summary table.

Examples

```
# using lm() for linear models
m1 <- lm(mpg ~ wt + hp, data = mtcars)
konfound(m1, wt)
konfound(m1, wt, to_return = "table")

# using glm() for non-linear models
if (requireNamespace("forcats")) {
  d <- forcats::gss_cat

  d$married <- ifelse(d$marital == "Married", 1, 0)

  m2 <- glm(married ~ age, data = d, family = binomial(link = "logit"))
  konfound(m2, age)
}

# using lme4 for mixed effects (or multi-level) models
if (requireNamespace("lme4")) {
  library(lme4)
  m3 <- fm1 <- lme4::lmer(Reaction ~ Days + (1 | Subject), sleepstudy)
  konfound(m3, Days)
}

m4 <- glm(outcome ~ condition, data = binary_dummy_data, family = binomial(link = "logit"))
konfound(m4, condition, two_by_two = TRUE, n_treat = 55)
```

`konfound_glm` *Konfound Analysis for Generalized Linear Models*

Description

This function performs konfound analysis on a generalized linear model object. It uses 'broom' to tidy model outputs and calculates the sensitivity of inferences. It supports analysis for a single variable or multiple variables.

Usage

```
konfound_glm(
  model_object,
  tested_variable_string,
  alpha,
  tails,
  index = "RIR",
  to_return
)
```

Arguments

`model_object`	The model object produced by glm.
`tested_variable_string`	The name of the variable being tested.
`alpha`	Significance level for hypothesis testing.
`tails`	Number of tails for the test (1 or 2).
`index`	Type of sensitivity analysis ('RIR' by default).
`to_return`	The type of output to return.

Value

The results of the konfound analysis for the specified variable(s).

`konfound_glm_dichotomous` *Konfound Analysis for Generalized Linear Models with Dichotomous Outcomes*

Description

This function performs konfound analysis on a generalized linear model object with a dichotomous outcome. It uses 'broom' to tidy model outputs and calculates the sensitivity of inferences.

Usage

```
konfound_glm_dichotomous(
  model_object,
  tested_variable_string,
  alpha,
  tails,
  to_return,
  n_treat,
  switch_trm,
  replace
)
```

Arguments

`model_object`	The model object produced by glm.
`tested_variable_string`	The name of the variable being tested.
`alpha`	Significance level for hypothesis testing.
`tails`	Number of tails for the test (1 or 2).
`to_return`	The type of output to return.
`n_treat`	Number of treatment cases.
`switch_trm`	Term to switch for sensitivity analysis.
`replace`	Boolean indicating whether to replace cases or not.

Value

The results of the konfound analysis.

`konfound_lm` *Konfound Analysis for Linear Models*

Description

This function performs konfound analysis on a linear model object produced by lm. It calculates the sensitivity of inferences for coefficients in the model. It supports analysis for a single variable or multiple variables.

Usage

```
konfound_lm(
  model_object,
  tested_variable_string,
  alpha,
  tails,
  index,
  to_return
)
```

Arguments

`model_object`	The linear model object produced by lm.
`tested_variable_string`	The name of the variable being tested.
`alpha`	Significance level for hypothesis testing.
`tails`	Number of tails for the test (1 or 2).
`index`	Type of sensitivity analysis ('RIR' by default).
`to_return`	The type of output to return.

Value

The results of the konfound analysis for the specified variable(s).

`konfound_lmer` *Konfound Analysis for Linear Mixed-Effects Models*

Description

This function performs konfound analysis on a linear mixed-effects model object produced by lme4::lmer. It calculates the sensitivity of inferences for fixed effects in the model. It supports analysis for a single variable or multiple variables.

Usage

```
konfound_lmer(
  model_object,
  tested_variable_string,
  test_all,
  alpha,
  tails,
  index,
  to_return
)
```

Arguments

`model_object`	The mixed-effects model object produced by lme4::lmer.
`tested_variable_string`	The name of the fixed effect being tested.
`test_all`	Boolean indicating whether to test all fixed effects or not.
`alpha`	Significance level for hypothesis testing.
`tails`	Number of tails for the test (1 or 2).
`index`	Type of sensitivity analysis ('RIR' by default).
`to_return`	The type of output to return.

Value

The results of the konfound analysis for the specified fixed effect(s).

`mkonfound` *Meta-Analysis and Sensitivity Analysis for Multiple Studies*

Description

Performs sensitivity analysis for multiple models, where parameters are stored in a data frame. It calculates the amount of bias required to invalidate or sustain an inference for each case in the data frame.

Usage

```
mkonfound(d, t, df, alpha = 0.05, tails = 2, return_plot = FALSE)
```

Arguments

`d`	A data frame or tibble containing t-statistics and associated degrees of freedom.
`t`	Column name or vector of t-statistics.
`df`	Column name or vector of degrees of freedom associated with t-statistics.
`alpha`	Significance level for hypothesis testing.
`tails`	Number of tails for the test (1 or 2).
`return_plot`	Whether to return a plot of the percent bias (default is ‘FALSE‘).

Value

Depending on ‘return_plot‘, either returns a data frame with analysis results or a plot.

Examples

```
## Not run:
mkonfound_ex
str(d)
mkonfound(mkonfound_ex, t, df)

## End(Not run)
```

`mkonfound_ex` *Example data for the mkonfound function*

Description

A dataset containing t and df values from example studies from Educational Evaluation and Policy Analysis (as detailed in Frank et al., 2013): https://drive.google.com/file/d/1aGhxGjvMvEPVAgOA8rrxvA97uUO5TTMe/vie

Usage

```
mkonfound_ex
```

Format

A data frame with 30 rows and 2 variables:

t t value

df degrees of freedom associated with the t value ...

Source

https://drive.google.com/file/d/1aGhxGjvMvEPVAgOA8rrxvA97uUO5TTMe/view

---

`output_df` *Output data frame based on model estimates and thresholds*

Description

Output data frame based on model estimates and thresholds

Usage

```
output_df(
  est_eff,
  beta_threshhold,
  unstd_beta,
  bias = NULL,
  sustain = NULL,
  recase,
  obs_r,
  critical_r,
  r_con,
  itcv,
  non_linear
)
```

Arguments

`est_eff`	estimated effect
`beta_threshhold`	threshold for beta
`unstd_beta`	unstandardized beta value
`bias`	bias to change inference
`sustain`	sustain to change inference
`recase`	number of cases to replace null
`obs_r`	observed correlation
`critical_r`	critical correlation
`r_con`	correlation for omitted variable
`itcv`	inferential threshold for confounding variable
`non_linear`	flag for non-linear models

Value

data frame with model information

`output_print` *Output printed text with formatting*

Description

This function outputs printed text for various indices such as RIR (Robustness of Inference to Replacement) and IT (Impact Threshold for a Confounding Variable) with specific formatting like bold, underline, and italic using functions from the crayon package. It handles different scenarios based on the effect difference, beta threshold, and other parameters, providing formatted output for each case.

Usage

```
output_print(
  n_covariates,
  est_eff,
  beta_threshhold,
  bias = NULL,
  sustain = NULL,
  nu,
  eff_thr,
  recase,
  obs_r,
  critical_r,
  r_con,
  itcv,
```

```
  alpha,
  index,
  far_bound,
  sdx = NA,
  sdy = NA,
  R2 = NA,
  rxcv = NA,
  rycv = NA,
  rxcvGz,
  rycvGz,
  benchmark_corr_product = NA,
  itcv_ratio_to_benchmark = NA
)
```

Arguments

`n_covariates`	number of covariates.
`est_eff`	The estimated effect.
`beta_threshhold`	The threshold value of beta, used for statistical significance determination.
`bias`	The percentage of the estimate that could be due to bias (optional).
`sustain`	The percentage of the estimate necessary to sustain an inference (optional).
`nu`	The hypothesized effect size used in replacement analysis.
`eff_thr`	Threshold for estimated effect.
`recase`	The number of cases that need to be replaced to change the inference.
`obs_r`	The observed correlation coefficient in the data.
`critical_r`	The critical correlation coefficient for statistical significance.
`r_con`	The correlation coefficient of an omitted variable with both the outcome and the predictor.
`itcv`	The impact threshold for a confounding variable.
`alpha`	The level of statistical significance.
`index`	A character string indicating the index for which the output is generated ('RIR' or 'IT').
`far_bound`	Indicator whether the threshold is towards the other side of nu or 0, by default is zero (same side), alternative is one (the other side).
`sdx`	Standard deviation of x.
`sdy`	Standard deviation of y.
`R2`	the unadjusted, original R2 in the observed function.
`rxcv`	the correlation between x and CV.
`rycv`	the correlation between y and CV.
`rxcvGz`	the correlation between predictor of interest and CV necessary to nullify the inference for smallest impact, conditioning on all observed covariates.

`rycvGz`	the correlation between outcome and CV necessary to nullify the inference for smallest impact, conditioning on all observed covariates.
`benchmark_corr_product`	the product of the correlations of covariates Z with X and Y (Rxz * Ryz), measuring the observed association strength.
`itcv_ratio_to_benchmark`	the ratio of the ITCV to the benchmark_corr_product, indicating the robustness of inference.

`output_table` *Output a Tidy Table from a Model Object*

Description

This function takes a model object and the tested variable, tidies the model output using ‘broom::tidy‘, calculates the impact threshold for confounding variables (ITCV) and impact for each covariate,and returns a rounded, tidy table of model outputs.

Usage

```
output_table(model_object, tested_variable)
```

Arguments

`model_object`	A model object from which to generate the output.
`tested_variable`	The variable being tested in the model.

Value

A tidy data frame containing model outputs, ITCV, and impacts for covariates.

`pkonfound` *Perform sensitivity analysis for published studies*

Description

For published studies, this command calculates (1) how much bias there must be in an estimate to nullify/sustain an inference; (2) the impact of an omitted variable necessary to nullify/sustain an inference for a regression coefficient. For a full description of the command’s usage and additional examples, please refer to our practical guide.

Usage

```
pkonfound(
  est_eff,
  std_err,
  n_obs,
  n_covariates = 1,
  alpha = 0.05,
  tails = 2,
  index = "RIR",
  nu = 0,
  n_treat = NULL,
  switch_trm = TRUE,
  model_type = "ols",
  a = NULL,
  b = NULL,
  c = NULL,
  d = NULL,
  two_by_two_table = NULL,
  test = "fisher",
  replace = "control",
  sdx = NA,
  sdy = NA,
  R2 = NA,
  far_bound = 0,
  eff_thr = NA,
  FR2max = 0,
  FR2max_multiplier = 1.3,
  to_return = "print",
  upper_bound = NULL,
  lower_bound = NULL,
  raw_treatment_success = NULL
)
```

Arguments

`est_eff`	the estimated effect (e.g., an unstandardized beta coefficient or a group mean difference).
`std_err`	the standard error of the estimate of the unstandardized regression coefficient.
`n_obs`	the number of observations in the sample.
`n_covariates`	the number of covariates in the regression model.
`alpha`	the probability of rejecting the null hypothesis (defaults to 0.05).
`tails`	integer indicating if the test is one-tailed (1) or two-tailed (2; defaults to 2).
`index`	specifies whether output is RIR or IT (impact threshold); defaults to `"RIR"`.
`nu`	specifies the hypothesis to be tested; defaults to testing whether `est_eff` is significantly different from 0.

`n_treat`	the number of cases associated with the treatment condition (for logistic regression models).
`switch_trm`	indicates whether to switch the treatment and control cases; defaults to `FALSE`.
`model_type`	the type of model; defaults to `"ols"`, but can be set to `"logistic"`.
`a`	the number of cases in the control group showing unsuccessful results (2x2 table model).
`b`	the number of cases in the control group showing successful results (2x2 table model).
`c`	the number of cases in the treatment group showing unsuccessful results (2x2 table model).
`d`	the number of cases in the treatment group showing successful results (2x2 table model).
`two_by_two_table`	a table (matrix, data.frame, tibble, etc.) from which `a`, `b`, `c`, and `d` can be extracted.
`test`	specifies whether to use Fisher's Exact Test (`"fisher"`) or a chi-square test (`"chisq"`); defaults to `"fisher"`.
`replace`	specifies whether to use the entire sample (`"entire"`) or the control group (`"control"`) for calculating the base rate; default is `"control"`.
`sdx`	the standard deviation of X (used for unconditional ITCV).
`sdy`	the standard deviation of Y (used for unconditional ITCV).
`R2`	the unadjusted, original R^2 in the observed function (used for unconditional ITCV).
`far_bound`	indicates whether the estimated effect is moved to the boundary closer (0, default) or further away (1).
`eff_thr`	for RIR: the unstandardized coefficient threshold to change an inference; for IT: the correlation defining the threshold for inference.
`FR2max`	the largest R^2 (or $R^2_{\max}$) in the final model with an unobserved confounder (used for COP).
`FR2max_multiplier`	the multiplier applied to R^2 to derive $R^2_{\max}$; defaults to 1.3 (used for COP).
`to_return`	specifies the output format: `"print"` (default) to display output, `"plot"` for a plot, or `"raw_output"` to return a data.frame for further analysis.
`upper_bound`	optional (replaces `est_eff`); the upper bound of the confidence interval.
`lower_bound`	optional (replaces `est_eff`); the lower bound of the confidence interval.
`raw_treatment_success`	optional; the unadjusted count of successful outcomes in the treatment group for calculating the specific RIR benchmark.

Details

The function accepts arguments depending on the type of model:

Linear Models (index: RIR, ITCV, PSE, COP)

- est_eff, std_err, n_obs, n_covariates, alpha, tails, index, nu
- sdx, sdy, R2, far_bound, eff_thr, FR2max, FR2max_multiplier
- upper_bound, lower_bound

Logistic Regression Model

- est_eff, std_err, n_obs, n_covariates, n_treat, alpha, tails, nu
- replace, switch_trm, raw_treatment_success, model_type

2x2 Table Model (Non-linear)

- a, b, c, d, two_by_two_table, test, replace, switch_trm

Values

pkonfound prints the bias and the number of cases that would have to be replaced with cases for which there is no effect to nullify the inference. If `to_return = "raw_output"`, a list is returned with the following components:

RIR & ITCV for linear model:

`obs_r` correlation between predictor of interest (X) and outcome (Y) in the sample data

`act_r` correlation between predictor of interest (X) and outcome (Y) from the sample regression based on the t-ratio accounting for non-zero null hypothesis

`critical_r` critical correlation value at which the inference would be nullified (e.g., associated with p=.05)

`r_final` final correlation value given confounding variable (CV). Should be equal to `critical_r`

`rxcv` unconditional $corr(X, CV)$ necessary to nullify the inference for smallest impact

`rycv` unconditional $corr(Y, CV)$ necessary to nullify the inference for smallest impact

`rxcvGz` $corr(X, CV|Z)$ conditioning on all observed covariates

`rycvGz` $corr(Y, CV|Z)$ conditioning on all observed covariates

`itcv` unconditional ITCV (`uncond_rxcv * uncond_rycv`)

`itcvGz` conditional ITCV given all observed covariates

`r2xz` R^2 using all observed covariates to explain the predictor of interest (X)

`r2yz` R^2 using all observed covariates to explain the predictor of interest (Y)

`beta_threshold` threshold for for estimated effect

`beta_threshold_verify` verified threshold matching `beta_threshold`

`perc_bias_to_change` percent bias to change inference

`RIR_primary` Robustness of Inference to Replacement (RIR)

`RIR_supplemental` RIR for an extra row or column that is needed to nullify the inference

`RIR_perc` RIR as % of total sample (for linear regression) or as % of data points in the cell where replacement takes place (for logistic and 2 by 2 table)

`Fig_ITCV` ITCV plot object

`Fig_RIR` RIR threshold plot object

COP for linear model:

`delta*` delta calculated using Oster's unrestricted estimator `delta*restricted` delta calculated using Oster's restricted estimator

`delta_exact` delta calculated using correlation-based approach

`delta_pctbias` percent bias when comparing `delta*` to `delta_exact`

`var(Y)` variance of the dependent variable (σ_Y^2)

`var(X)` variance of the independent variable (σ_X^2)

`var(CV)` variance of the confounding variable (σ_{CV}^2)

`cor_oster` correlation matrix implied by `delta*`

`cor_exact` correlation matrix implied by `delta_exact`

`eff_x_M3_oster` effect estimate for X under the Oster-PSE variant

`eff_x_M3` effect estimate for X under the PSE adjustment

`Table` formatted results table

`Figure` COP diagnostic plot

PSE for linear model:

`corr(X,CV|Z)` correlation between X and CV conditional on Z

`corr(Y,CV|Z)` correlation between Y and CV conditional on Z

`corr(X,CV)` correlation between X and CV

`corr(Y,CV)` correlation between X and CV

`covariance matrix` covariance matrix among Y, X, Z, and CV under the PSE adjustment

`eff_M3` estimated unstandardized regression coefficient for X in M3 under the PSE adjustment

`se_M3` standard error of that coefficient in M3 under the PSE adjustment

`Table` matrix summarizing key statistics from three nested regression models (M1, M2, M3)

RIR for logistic model:

`RIR_primary` Robustness of Inference to Replacement (RIR)

`RIR_supplemental` RIR for an extra row or column that is needed to nullify the inference

`RIR_perc` RIR as % of data points in the cell where replacement takes place

`fragility_primary` Fragility; the number of switches (e.g., treatment success to treatment failure) to nullify the inference

`fragility_supplemental` Fragility for an extra row or column that is needed to nullify the inference

`starting_table` observed (implied) 2 by 2 table before replacement and switching

`final_table` the 2 by 2 table after replacement and switching

`user_SE` user-entered standard error

`analysis_SE` the standard error used to generate a plausible 2 by 2 table

`needtworows` indicator whether extra switches were needed

RIR for 2×2 table model:

`RIR_primary` Robustness of Inference to Replacement (RIR)

`RIR_supplemental` RIR for an extra row or column that is needed to nullify the inference

`RIR_perc` RIR as % of data points in the cell where replacement takes place

`fragility_primary` Fragility; the number of switches (e.g., treatment success to treatment failure) to nullify the inference `fragility_supplemental` Fragility for an extra row or column that is needed to nullify the inference

`starting_table` observed 2 by 2 table before replacement and switching

`final_table` the 2 by 2 table after replacement and switching

`needtworows` indicator whether extra switches were needed

Note

For a thoughtful background on benchmark options for ITCV, see doi:10.1111/rssb.12348 (Cinelli & Hazlett, 2020), doi:10.1177/01492063241293126 (Lonati & Wulff, 2024), and doi:10.1177/0049124100029002001 (Frank, 2000).

Examples

```
## Linear models
pkonfound(2, .4, 100, 3)
pkonfound(-2.2, .65, 200, 3)
pkonfound(.5, 3, 200, 3)
pkonfound(-0.2, 0.103, 20888, 3, n_treat = 17888, model_type = "logistic")

# using a confidence interval
pkonfound(upper_bound = 3, lower_bound = 1, n_obs = 100, n_covariates = 3)

pkonfound(2, .4, 100, 3, to_return = "thresh_plot")
pkonfound(2, .4, 100, 3, to_return = "corr_plot")

## Logistic regression model example
pkonfound(-0.2, 0.103, 20888, 3, n_treat = 17888, model_type = "logistic")

## 2x2 table examples
pkonfound(a = 35, b = 17, c = 17, d = 38)
pkonfound(a = 35, b = 17, c = 17, d = 38, alpha = 0.01)
pkonfound(a = 35, b = 17, c = 17, d = 38, alpha = 0.01, switch_trm = FALSE)
pkonfound(a = 35, b = 17, c = 17, d = 38, test = "chisq")

## Advanced examples
# Calculating unconditional ITCV and benchmark correlation for ITCV
pkonfound(est_eff = .5, std_err = .056, n_obs = 6174, sdx = 0.22, sdy = 1, R2 = .3,
          index = "IT", to_return = "print")
# Calculating delta* and delta_exact
pkonfound(est_eff = .4, std_err = .1, n_obs = 290, sdx = 2, sdy = 6, R2 = .7,
         eff_thr = 0, FR2max = .8, index = "COP", to_return = "raw_output")
# Calculating rxcv and rycv when preserving standard error
pkonfound(est_eff = .5, std_err = .056, n_obs = 6174, eff_thr = .1,
         sdx = 0.22, sdy = 1, R2 = .3, index = "PSE", to_return = "raw_output")
```

`plot_correlation` *Plot Correlation Diagram*

Description

This function creates a plot to illustrate the correlation between different variables,specifically focusing on the confounding variable, predictor of interest, and outcome.It uses ggplot2 for graphical representation.

Usage

```
plot_correlation(r_con, obs_r, critical_r)
```

Arguments

`r_con`	Correlation coefficient related to the confounding variable.
`obs_r`	Observed correlation coefficient.
`critical_r`	Critical correlation coefficient for decision-making.

Value

A ggplot object representing the correlation diagram.

`plot_threshold` *Plot Effect Threshold Diagram*

Description

This function creates a plot to illustrate the threshold of an effect estimate in relation to a specified beta threshold. It uses ggplot2 for graphical representation.

Usage

```
plot_threshold(beta_threshold, est_eff)
```

Arguments

`beta_threshold`	The threshold value for the effect.
`est_eff`	The estimated effect size.

Value

A ggplot object representing the effect threshold diagram.

`tkonfound` *Perform Sensitivity Analysis on 2x2 Tables*

Description

This function performs a sensitivity analysis on a 2x2 contingency table. It calculates the number of cases that need to be replaced to invalidate or sustain the statistical inference. The function also allows switching between treatment success and failure or control success and failure based on the provided parameters.

Usage

```
tkonfound(
  a,
  b,
  c,
  d,
  alpha = 0.05,
  switch_trm = TRUE,
  test = "fisher",
  replace = "control",
  to_return = to_return
)
```

Arguments

`a`	Number of unsuccessful cases in the control group.
`b`	Number of successful cases in the control group.
`c`	Number of unsuccessful cases in the treatment group.
`d`	Number of successful cases in the treatment group.
`alpha`	Significance level for the statistical test, default is 0.05.
`switch_trm`	Boolean indicating whether to switch treatment row cells, default is TRUE.
`test`	Type of statistical test to use, either "fisher" (default) or "chisq".
`replace`	Indicates whether to use the entire sample or the control group for base rate calculation, default is "control".
`to_return`	Type of output to return, either "raw_output" or "print".

Value

Returns detailed information about the sensitivity analysis, including the number of cases to be replaced (RIR), user-entered table, transfer table, and conclusions.

`tkonfound_fig` *Draw Figures for Change in Effect Size in 2x2 Tables*

Description

This function generates plots illustrating how the change in effect size is influenced by switching or replacing outcomes in a 2x2 table. It produces two plots: one showing all possibilities (switching) and another zoomed in the area for positive RIR (Relative Impact Ratio).

Usage

```
tkonfound_fig(
  a,
  b,
  c,
  d,
  thr_p = 0.05,
  switch_trm = TRUE,
  test = "fisher",
  replace = "control"
)
```

Arguments

`a`	Number of cases in the control group with unsuccessful outcomes.
`b`	Number of cases in the control group with successful outcomes.
`c`	Number of cases in the treatment group with unsuccessful outcomes.
`d`	Number of cases in the treatment group with successful outcomes.
`thr_p`	P-value threshold for statistical significance, default is 0.05.
`switch_trm`	Whether to switch the two cells in the treatment or control row, default is TRUE (treatment row).
`test`	Type of statistical test used, either "Fisher's Exact Test" (default) or "Chi-square test".
`replace`	Indicates whether to use the entire sample or just the control group for calculating the base rate, default is "control".

Value

Returns two plots showing the effect of hypothetical case switches on the effect size in a 2x2 table.

Examples

```
tkonfound_fig(14, 17, 6, 25, test = "chisq")
```

`verify_reg_Gzcv` *Verify regression model with control variable Z*

Description

Verify regression model with control variable Z

Usage

```
verify_reg_Gzcv(n_obs, sdx, sdy, sdz, sdcv, rxy, rxz, rzy, rcvy, rcvx, rcvz)
```

Arguments

`n_obs`	number of observations
`sdx`	standard deviation of X
`sdy`	standard deviation of Y
`sdz`	standard deviation of Z
`sdcv`	sd between C and V
`rxy`	correlation coefficient between X and Y
`rxz`	correlation coefficient between X and Z
`rzy`	correlation coefficient between Z and Y
`rcvy`	correlation coefficient between V and Y
`rcvx`	correlation coefficient between V and X
`rcvz`	correlation coefficient between V and Z

Value

list of model parameters

`verify_reg_uncond` *Verify unconditional regression model*

Description

Verify unconditional regression model

Usage

```
verify_reg_uncond(n_obs, sdx, sdy, rxy)
```

Arguments

`n_obs`	number of observations
`sdx`	standard deviation of X
`sdy`	standard deviation of Y
`rxy`	correlation coefficient between X and Y

Value

list of model parameters

`zzz` *Package Initialization Functions and Utilities*

Description

These functions are used for initializing the package environment and providing utility functions for the package.

Index