

Package ‘konfound’

December 17, 2020

Type Package

Title Quantify the Robustness of Causal Inferences

Version 0.3.0

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 α with the predictor of interest (e.g., treatment) and outcome to invalidate an inference of a treatment effect”. Or “one would have to replace p percent of the observed data with null hypothesis cases 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.

License MIT + file LICENSE

Imports broom, broom.mixed, crayon, dplyr, ggplot2, mice, purrr, rlang, tidyr, tibble

Suggests margins, pbkrtest, devtools, forcats, knitr, lme4, rmarkdown, roxygen2, testthat, ggrepel, covr

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

URL <https://github.com/jrosen48/konfound>

BugReports <https://github.com/jrosen48/konfound/issues>

Depends R (>= 2.10)

NeedsCompilation no

Author Joshua M Rosenberg [aut, cre],
 Ran Xu [ctb],
 Qinyun Lin [ctb],
 Spiro Maroulis [ctb],
 Kenneth A Frank [ctb]

Maintainer Joshua M Rosenberg <jmichaelrosenberg@gmail.com>

Repository CRAN

Date/Publication 2020-12-17 17:30:06 UTC

R topics documented:

binary_dummy_data	2
concord1	2
konfound	3
launch_shiny	4
mkonfound	5
pkonfound	6
summary.konfound	8

Index	9
--------------	----------

binary_dummy_data	<i>Binary dummy data</i>
-------------------	--------------------------

Description

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

Format

A data.frame with 107 rows and 2 variables.

concord1	<i>Concord1 data</i>
----------	----------------------

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.

konfound	<i>Perform sensitivity analysis on fitted models</i>
----------	--

Description

For fitted models, this command calculates (1) how much bias there must be in an estimate to invalidate/sustain an inference; (2) the impact of an omitted variable necessary to invalidate/sustain an inference for a regression coefficient. Currently works for: models created with `lm()` (linear models).

Usage

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

Arguments

<code>model_object</code>	output from a model (currently works for: <code>lm</code>)
<code>tested_variable</code>	Variable associated with the unstandardized beta coefficient to be tested
<code>alpha</code>	probability of rejecting the null hypothesis (defaults to 0.05)
<code>tails</code>	integer whether hypothesis testing is one-tailed (1) or two-tailed (2; defaults to 2)
<code>index</code>	whether output is RIR or IT (impact threshold); defaults to "RIR"
<code>to_return</code>	whether to return a <code>data.frame</code> (by specifying this argument to equal "raw_output" for use in other analyses) or a plot ("plot"); default is to print ("print") the output to the console; can specify a vector of output to return
<code>test_all</code>	whether to carry out the sensitivity test for all of the coefficients (defaults to FALSE)

two_by_two	whether or not the tested variable is a dichotomous variable in a GLM; if so, the 2X2 table approach is used; only works for single variables at present (so test_all = TRUE will return an error)
n_treat	the number of cases associated with the treatment condition; applicable only when model_type = "logistic"
switch_trm	whether to switch the treatment and control cases; defaults to FALSE; applicable only when model_type = "logistic"
replace	whether using entire sample or the control group to calculate the base rate; default is the entire sample

Value

prints the bias and the number of cases that would have to be replaced with cases for which there is no effect to invalidate the inference

Examples

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

Description

Open interactive web application for konfound

Usage

```
launch_shiny()
```

Details

Open the Shiny interactive web application in a browser

Value

Launches a web browser

mkonfound	<i>Perform meta-analyses including sensitivity analysis</i>
-----------	---

Description

For fitted models, this command carries out sensitivity analysis for a number of models, when their parameters stored in a data.frame.

Usage

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

Arguments

d	data.frame or tibble with the t-statistics and associated degrees of freedom
t	t-statistic or vector of t-statistics
df	degrees of freedom or vector of degrees of freedom associated with the t-statistics in the t argument
alpha	probability of rejecting the null hypothesis (defaults to 0.05)
tails	integer whether hypothesis testing is one-tailed (1) or two-tailed (2; defaults to 2)
return_plot	whether to return a plot of the percent bias; defaults to FALSE

Value

prints the bias and the number of cases that would have to be replaced with cases for which there is no effect to invalidate the inference for each of the cases in the data.frame

Examples

```
d <- read.csv("https://msu.edu/~kenfrank/example%20dataset%20for%20mkonfound.csv")
d <- d[1:3, ] # this is only so that the example runs more quickly
str(d)
mkonfound(d, t, df)
```

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 invalidate/sustain an inference; (2) the impact of an omitted variable necessary to invalidate/sustain an inference for a regression coefficient.

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",
  to_return = "print"
)
```

Arguments

est_eff	the estimated effect (such as 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	probability of rejecting the null hypothesis (defaults to 0.05)
tails	integer whether hypothesis testing is one-tailed (1) or two-tailed (2; defaults to 2)
index	whether output is RIR or IT (impact threshold); defaults to "RIR"
nu	what 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; applicable only when model_type = "logistic"
switch_trm	whether to switch the treatment and control cases; defaults to FALSE; applicable only when model_type = "logistic"
model_type	the type of model being estimated; defaults to "ols" for a linear regression model; the other option is "logistic"
a	cell is the number of cases in the control group showing unsuccessful results
b	cell is the number of cases in the control group showing successful results
c	cell is the number of cases in the treatment group showing unsuccessful results
d	cell is the number of cases in the treatment group showing successful results
two_by_two_table	table that is a matrix or can be coerced to one (data.frame, tibble, tribble) from which the a, b, c, and d arguments can be extracted
test	whether using Fisher's Exact Test or A chi-square test; defaults to Fisher's Exact Test
replace	whether using entire sample or the control group to calculate the base rate; default is the entire sample
to_return	whether to return a data.frame (by specifying this argument to equal "raw_output" for use in other analyses) or a plot ("plot"); default is to print ("print") the output to the console; can specify a vector of output to return

Value

prints the bias and the number of cases that would have to be replaced with cases for which there is no effect to invalidate the inference

Examples

```
# using pkonfound for 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")

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

pkonfound_output <- pkonfound(2, .4, 200, 3,
  to_return = c("raw_output", "thresh_plot", "corr_plot")
)
```

```
summary(pkonfound_output)
pkonfound_output$raw_output
pkonfound_output$thresh_plot
pkonfound_output$corr_plot

# using pkonfound for a 2x2 table
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")
```

summary.konfound *Concise summary of konfound output*

Description

Concise summary of konfound output

Usage

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

Arguments

object	A 'konfound' object
...	Additional arguments

Details

Prints a concise summary of konfound output with multiple types of data specified in the to_return argument

Index

[binary_dummy_data](#), [2](#)

[concord1](#), [2](#)

[konfound](#), [3](#)

[launch_shiny](#), [4](#)

[mkonfound](#), [5](#)

[pkonfound](#), [6](#)

[summary.konfound](#), [8](#)