

Package ‘konfound’

April 6, 2018

Type Package

Title Quantify the Robustness of Causal Inferences

Version 0.1.0

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, dplyr, ggplot2, margins, pbkrtest, purrr, rlang, tidyr

Suggests devtools, forcats, knitr, lme4, rmarkdown, roxygen2, testthat

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Author Joshua M Rosenberg [aut, cre],

Ran Xu [ctb],

Kenneth A Frank [ctb]

Maintainer Joshua M Rosenberg <jmichaelrosenberg@gmail.com>

Repository CRAN

Date/Publication 2018-04-06 16:12:41 UTC

R topics documented:

konfound	2
launch_shiny	3
mkonfound	4
pkonfound	4
summary.konfound	6

Index	7
--------------	----------

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,
         to_return = "print", test_all = FALSE)
```

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>to_return</code>	whether to return a <code>data.frame</code> (by specifying this argument to equal <code>"raw_output"</code> for use in other analyses) or a plot (<code>"plot"</code>); default is to print (<code>"print"</code>) 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 <code>FALSE</code>)

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)
}
```

launch_shiny

Open interactive web application for konfound

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

Perform meta-analyses including sensitivity analysis

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")
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, nu = 0, to_return = "print")
```

Arguments

<code>est_eff</code>	the estimated effect (such as an unstandardized beta coefficient or a group mean difference)
<code>std_err</code>	the standard error of the estimate of the unstandardized regression coefficient
<code>n_obs</code>	the number of observations in the sample
<code>n_covariates</code>	the number of covariates in the regression model
<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>nu</code>	what hypothesis to be tested; defaults to testing whether <code>est_eff</code> is significantly different from 0
<code>to_return</code>	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

```
pkonfound(2, .4, 100, 3)
pkonfound(-2.2, .65, 200, 3)
pkonfound(.5, 3, 200, 3)

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
```

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

[konfound](#), [2](#)

[launch_shiny](#), [3](#)

[mkonfound](#), [4](#)

[pkonfound](#), [4](#)

[summary.konfound](#), [6](#)