Package ‘pvaluefunctions’

November 29, 2019

Type Package

Title Creates and Plots P-Value Functions, S-Value Functions, Confidence Distributions and Confidence Densities

Version 1.5.0

Maintainer Denis Infanger <denis.infanger@unibas.ch>


Depends R (>= 3.1.0)

Imports ggplot2 (>= 3.2.1), scales (>= 1.1.0), zipfR (>= 0.6-66), stats

Suggests knitr, rmarkdown, devtools

License GPL-3

URL https://github.com/DInfanger/pvaluefunctions

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.0.1

NeedsCompilation no

Author Denis Infanger [aut, cre] (<https://orcid.org/0000-0001-9028-7110>)

Repository CRAN

Date/Publication 2019-11-29 19:30:02 UTC
R topics documented:

conf_dist

Description

The function `conf_dist` generates confidence distributions (cdf), confidence densities (pdf), Shannon suprasal (s-value) functions and \( p \)-value functions for several commonly used estimates. In addition, counternulls (see Rosenthal et al. 1994) and point estimates are calculated.

Usage

```r
conf_dist(
  estimate = NULL,
  n = NULL,
  df = NULL,
  stderr = NULL,
  tstat = NULL,
  type = NULL,
  plot_type = c("p_val", "s_val", "cdf", "pdf"),
  n_values = 10000L,
  est_names = NULL,
  conf_level = NULL,
  null_values = NULL,
  trans = "identity",
  alternative = c("two_sided", "one_sided"),
  log_yaxis = FALSE,
  cut_logyaxis = 0.05,
  xlab = NULL,
  xlim = NULL,
  together = FALSE,
  nrow = NULL,
  ncol = NULL,
  plot_p_limit = (1 - 0.999),
  plot_counternull = FALSE,
  title = NULL,
  ylab = NULL,
  ylab_sec = NULL,
  inverted = FALSE,
  x_scale = c("default", "linear", "logarithm"),
  plot = TRUE
)
```
Arguments

estimate  Numerical vector containing the estimate(s).

n  Numerical vector containing the sample size(s). Required for correlations, variances, proportions and differences between proportions. Must be equal the number of estimates.

df  Numerical vector containing the degrees of freedom. Required for statistics based on the t-distribution (e.g. linear regression) and t-tests. Must be equal the number of estimates.

stderr  Numerical vector containing the standard error(s) of the estimate(s). Required for statistics based on the t-distribution (e.g. linear regression) and the normal distribution (e.g. logistic regression). Must be equal the number of estimate(s).

tstat  Numerical vector containing the t-statistic(s). Required for t-tests (means and mean differences). Must be equal the number of estimates.

type  String indicating the type of the estimate. Must be one of the following: ttest, linreg, gammareg, general_t, logreg, poisreg, coxreg, general_z, pearson, spearman, kendall, var, prop, propdiff.

plot_type  String indicating the type of plot. Must be one of the following: cdf (confidence distribution), pdf (confidence density), p_val (p-value function, the default), s_val (Surprisal value functions). For differences between independent proportions, only p-value functions and Surprisal values are available.

n_values  (optional) Integer indicating the number of points that are used to generate the graphics. The higher this number, the higher the computation time and resolution.

est_names  (optional) String vector indicating the names of the estimate(s). Must be equal the number of estimates.

conf_level  (optional) Numerical vector indicating the confidence level(s). Must be between 0 and 1.

null_values  (optional) Numerical vector indicating the null value(s) in the plot on the untransformed (original) scale. For example: The null values for an odds ratio of 1 is 0 on the log-odds scale. If x limits are specified with xlim, all null values outside of the specified x limits are ignored for plotting and a message is printed.

trans  (optional) String indicating the transformation function that will be applied to the estimates and confidence curves. For example: \texttt{exp} for an exponential transformation of the log-odds in logistic regression. Can be a custom function.

alternative  String indicating if the confidence level(s) are two-sided or one-sided. Must be one of the following: two_sided, one_sided.

log_yaxis  Logical. Indicating if a portion of the y-axis should be displayed on the logarithmic scale.

cut_logyaxis  Numerical value indicating the threshold below which the y-axis will be displayed logarithmically. Must lie between 0 and 1.

xlab  (optional) String indicating the label of the x-axis.

xlim  (optional) Optional numerical vector of length 2 (x1, x2) indicating the limits of the x-axis on the untransformed scale if \texttt{trans} is not \texttt{identity}. The scale of the
x-axis set by x_scale does not affect the x limits. For example: If you want to plot p-value functions for odds ratios from logistic regressions, the limits have to be given on the log-odds scale if trans = "exp". Note that x1 > x2 is allowed but then x2 will be the left limit and x1 the right limit (i.e. the limits are sorted before plotting). Null values (specified in null_values) that are outside of the specified limits are ignored and a message is printed.

### together
Logical. Indicating if graphics for multiple estimates should be displayed together or on separate plots.

### nrow
(optional) Integer greater than 0 indicating the number of rows when together = FALSE is specified for multiple estimates. Used in facet_wrap in ggplot2.

### ncol
(optional) Integer greater than 0 indicating the number of columns when together = FALSE is specified for multiple estimates. Used in facet_wrap in ggplot2.

### plot_p_limit
Numerical value indicating the lower limit of the y-axis. Must be greater than 0 for a logarithmic scale (i.e. \( \log_yaxis = \text{TRUE} \)). The default is to omit plotting p-values smaller than \( 1 - 0.999 = 0.001 \).

### plot_counternull
Logical. Indicating if the counternull should be plotted as a point. Only available for p-value functions and s-value functions. Counternull values that are outside of the plotted functions are not shown.

### title
(optional) String containing a title of the plot.

### ylab
(optional) String indicating the title for the primary (left) y-axis.

### ylab_sec
(optional) String indicating the title for the secondary (right) y-axis.

### inverted
Logical. Indicating the orientation of the y-axis for the P-value function (p_val), S-value function (s_val) and the confidence distribution (cdf). By default (i.e. inverted = FALSE) small P-values are plotted at the bottom and large ones at the top so that the cusp of the P-value function is at the top. By setting inverted = TRUE, the y-axis is inverted. Ignored for confidence densities.

### x_scale
String indicating the scaling of the x-axis. The default is to scale the x-axis logarithmically if the transformation specified in trans is "exp" (exponential) and linearly otherwise. The option linear (can be abbreviated) forces a linear scaling and the option logarithm (can be abbreviated) forces a logarithmic scaling, regardless what has been specified in trans.

### plot
Logical. Should a plot be created (TRUE, the default) or not (FALSE). FALSE can be useful if users want to create their own plots using the returned data from the function. If FALSE, no ggplot2 object is returned.

### Details

\( P \)-value functions and confidence intervals are calculated based on the \( t \)-distribution for \( t \)-tests, linear regression coefficients, and gamma regression models (GLM). The normal distribution is used for logistic regression, poisson regression and cox regression models. For correlation coefficients, Fisher’s transform is used using the corresponding variances (see Bonett et al. 2000). \( P \)-value functions and confidence intervals for variances are constructed using the Chi2 distribution. Finally, Wilson’s score intervals are used for one proportion. For differences of proportions, the Wilson score interval with continuity correction is used (Newcombe 1998).
conf_dist

Value

conf_dist returns four data frames and if plot = TRUE was specified, a ggplot2-plot object: res_frame (contains parameter values (e.g. mean differences, odds ratios etc.), p-values (one- and two-sided), s-values, confidence distributions and densities, variable names and type of hypothesis), conf_frame (contains the specified confidence level(s) and the corresponding lower and upper limits as well as the corresponding variable name), counternull_frame (contains the counternull and the corresponding null values), point_est (contains the mean, median and mode point estimates) and if plot = TRUE was specified, plot (a ggplot2 object).

References


See Also

 ggconcurve

Examples

#======================================================================================
# Create a p-value function for an estimate using the normal distribution
#======================================================================================
res <- conf_dist(
  estimate = c(-0.13),
  stderr = c(0.224494),
  type = "general_z"
)
conf_dist

, plot_type = "p_val"
, n_values = 1e4L
, est_names = c("Parameter value")
, log_yaxis = FALSE
, cut_logyaxis = 0.05
, conf_level = c(0.95)
, null_values = c(0)
, trans = "identity"
, alternative = "two_sided"
, xlab = "Var"
, xlim = c(-1, 1)
, together = TRUE
, plot_p_limit = 1 - 0.9999
, plot_counternull = TRUE
, title = NULL
, ylab = NULL
, ylab_sec = NULL
, inverted = FALSE
, x_scale = "default"
, plot = TRUE
)

#======================================================================================
# P-value function for a single regression coefficient (Agriculture in the model below)
#======================================================================================
mod <- lm(Infant.Mortality~Agriculture + Fertility + Examination, data = swiss)
summary(mod)
res <- conf_dist(
    estimate = c(-0.02143)
, df = c(43)
, stderr = (0.02394)
, type = "linreg"
, plot_type = "p_val"
, n_values = 1e4L
, conf_level = c(0.95, 0.90, 0.80)
, null_values = c(0)
, trans = "identity"
, alternative = "two_sided"
, log_yaxis = TRUE
, cut_logyaxis = 0.05
, xlab = "Coefficient Agriculture"
, together = FALSE
, plot_p_limit = 1 - 0.9999
, plot_counternull = FALSE
, title = NULL
, ylab = NULL
, ylab_sec = NULL
, inverted = FALSE
, x_scale = "default"
, plot = TRUE
)
# P-value function for an odds ratio (logistic regression), plotted with inverted y-axis

```r
res <- conf_dist(
  estimate = c(0.804037549),
  stderr = c(0.331819298),
  type = "logreg",
  plot_type = "p_val",
  n_values = 1e4L,
  est_names = c("GPA"),
  conf_level = c(0.95, 0.90, 0.80),
  null_values = c(log(1)) # null value on the log-odds scale
)
```

# Difference between two independent proportions: Newcombe with continuity correction

```r
res <- conf_dist(
  estimate = c(68/100, 98/150),
  n = c(100, 150),
  type = "propdiff",
  plot_type = "p_val",
  n_values = 1e4L,
  conf_level = c(0.95, 0.90, 0.80),
  null_values = c(0),
  trans = "identity",
  alternative = "two_sided",
  log_yaxis = FALSE,
  cut_logyaxis = 0.05,
  xlab = "Difference between proportions",
  together = FALSE,
  plot_p_limit = 1 - 0.9999,
  plot_counternull = FALSE,
  title = NULL
)
```
# Difference between two independent proportions: Agresti & Caffo

# First proportion
x1 <- 8
n1 <- 40

# Second proportion
x2 <- 11
n2 <- 30

# Apply the correction
p1hat <- (x1 + 1)/(n1 + 2)
p2hat <- (x2 + 1)/(n2 + 2)

# The original estimator
est0 <- (x1/n1) - (x2/n2)

# The unmodified estimator and its standard error using the correction
est <- p1hat - p2hat
se <- sqrt(((p1hat*(1 - p1hat))/(n1 + 2)) + ((p2hat*(1 - p2hat))/(n2 + 2)))

res <- conf_dist(
  estimate = c(est),
  stderr = c(se),
  type = "general_z",
  plot_type = "p_val",
  n_values = 1e4L,
  log_yaxis = FALSE,
  cut_logyaxis = 0.05,
  conf_level = c(0.95, 0.99),
  null_values = c(0, 0.3),
  trans = "identity",
  alternative = "two_sided",
  xlab = "Difference of proportions",
  together = FALSE,
  plot_p_limit = 1 - 0.9999,
  plot_counternull = FALSE,
  title = "P-value function for the difference of two independent proportions",
  ylab = NULL,
  ylab_sec = NULL,
  inverted = FALSE,
  x_scale = "default",
  plot = TRUE
)
# P-value function and confidence distribution for the relative survival effect (1 - HR%)
# Replicating Figure 1 in Bender et al. (2005)

# Define the transformation function and its inverse for the relative survival effect

rse_fun <- function(x){  # x is the log-hazard ratio
  100*(1 - exp(x))
}

rse_fun_inv <- function(x){
  log(1 - (x/100))
}

res <- conf_dist(
  estimate = log(0.72),
  stderr = 0.187618,
  type = "coxreg",
  plot_type = "p_val",
  n_values = 1e4L,
  est_names = c("RSE"),
  conf_level = c(0.95, 0.8, 0.5),
  null_values = rse_fun_inv(0),
  trans = "rse_fun",
  alternative = "two_sided",
  log_yaxis = FALSE,
  cut_logyaxis = 0.05,
  xlab = "Relative survival effect (1 - HR%)",
  xlim = rse_fun_inv(c(-30, 60)),
  together = FALSE,
  plot_p_limit = 1 - 0.999,
  plot_counternull = TRUE,
  inverted = TRUE,
  title = "Figure 1 in Bender et al. (2005)",
  x_scale = "default",
  plot = TRUE)
Index

conf_dist, 2

ggconcurve, 5