Package 'pvaluefunctions'

October 14, 2022

Type Package

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

Version 1.6.2

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

Description Contains functions to compute and plot confidence distributions, confidence densities, p-value functions and s-value (surprisal) functions for several commonly used estimates. Instead of just calculating one p-value and one confidence interval, p-value functions display p-values and confidence intervals for many levels thereby allowing to gauge the compatibility of several parameter values with the data. These methods are discussed by Infanger D, Schmidt-Trucksäss A. (2019) <doi:10.1002/sim.8293>; Poole C. (1987) <doi:10.2105/AJPH.77.2.195>; Schweder T, Hjort NL. (200 9469.00285>; Bender R, Berg G, Zeeb H. (2005) <doi:10.1002/bimj.200410104> ; Singh K, Xie M, Strawderman WE. (2007) <doi:10.1214/074921707000000102>; Rothman KJ, Greenland S, Lash TL. (2008, ISBN:9781451190052); Amrhein V, Trafimow D, Greenland S. (2019) <doi:10.1080/00031305.2018.1543137>; Greenland S. (2019) <doi:10.1080/00031305.2018.1529625> and Rafi Z, Greenland S. (2020) <doi:10.1186/s12874-020-01105-9>.

Depends R (>= 3.5.0)

Imports ggplot2 (>= 3.3.0), scales (>= 1.1.1), zipfR (>= 0.6-70),
pracma (>= 2.3.3), gsl (>= 2.1-7.1), stats

Suggests knitr, rmarkdown, devtools

License GPL-3

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

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.1.2

NeedsCompilation no

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

Repository CRAN

Date/Publication 2021-12-01 00:40:05 UTC

R topics documented:

---

conf_dist	*Create and Plot* P-*Value Functions, S-Value Functions, Confidence Distributions and Confidence Densities*

Description

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

Usage

```
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,
  plot_legend = TRUE,
  same_color = FALSE,
  col = "black",
  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 containqning 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). Bust 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: "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 trans is not 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.
plot_legend	Logical. Indicating if a legend should be plotted if multiple curves are plotted together with different colors (i.e. together = TRUE) and same_color = FALSE).
same_color	Logical. Indicating if curves should be distinguished using colors if they are plotted together (i.e. together = TRUE). Setting this to FALSE also disables the default behavior that the two halves of the curves are plotted in different colors for a one-sided alternative.
col	String indicating the colour of the curves. Only relevant for single curves, multiple curves not plotted together (i.e. together = FALSE) and multiple curves plotted together but with the option same_color set to TRUE.
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 = 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 a 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).

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, aucc_frame contains the estimated AUCC (area under the confidence curve, see Berrar 2017) calculated by trapezoidal integration on the untransformed scale. Also provides the proportion of the aucc that lies above the null value(s) if they are provided. plot (a ggplot2 object).

References

Bender R, Berg G, Zeeb H. Tutorial: using confidence curves in medical research. *Biom J.* 2005;47(2):237-247.

Berrar D. Confidence curves: an alternative to null hypothesis significance testing for the comparison of classifiers. *Mach Learn.* 2017;106:911-949.

Bonett DG, Wright TA. Sample size requirements for estimating Pearson, Kendall and Spearman correlations. *Psychometrika.* 2000;65(1):23-28.

Cole SR, Edwards JK, Greenland S. Surprise! *Am J Epidemiol.* 2021:190(2):191-193.

Infanger D, Schmidt-Trucksäss A. *P* value functions: An underused method to present research results and to promote quantitative reasoning. *Stat Med.* 2019;38:4189-4197.

Newcombe RG. Interval estimation for the difference between independent proportions: comparison of eleven methods. *Stat Med.* 1998;17:873-890.

Poole C. Confidence intervals exclude nothing. *Am J Public Health.* 1987;77(4):492-493.

Poole C. Beyond the confidence interval. *Am J Public Health.* 1987;77(2):195-199.

Rafi Z, Greenland S. Semantic and cognitive tools to aid statistical science: replace confidence and significance by compatibility and surprise. *BMC Med Res Methodol* 2020;20:244.

Rosenthal R, Rubin D. The counternull value of an effect size: a new statistic. *Psychological Science.* 1994;5(6):329-334.

Rothman KJ, Greenland S, Lash TL. Modern epidemiology. 3rd ed. Philadelphia, PA: Wolters Kluwer; 2008.

Schweder T, Hjort NL. Confidence, likelihood, probability: statistical inference with confidence distributions. New York, NY: Cambridge University Press; 2016.

Sullivan KM, Foster DA. Use of the confidence interval function. *Epidemiology.* 1990;1(1):39-42.

Xie Mg, Singh K. Confidence distribution, the frequentist distribution estimator of a parameter: A review. *Internat Statist Rev.* 2013;81(1):3-39.

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"
  , 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 - 0.999
    , 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
#===============================================================================

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
  , trans = "exp"
  , alternative = "two_sided"
  , log_yaxis = FALSE
  , cut_logyaxis = 0.05
  , xlab = "Odds Ratio (GPA)"
  , xlim = log(c(0.7, 5.2)) # axis limits on the log-odds scale
  , together = FALSE
  , plot_p_limit = 1 - 0.999
  , plot_counternull = TRUE
  , title = NULL
  , ylab = NULL
  , ylab_sec = NULL
  , inverted = TRUE
  , x_scale = "default"
  , plot = TRUE
)

#===============================================================================
```

```
# Difference between two independent proportions: Newcombe with continuity correction
#=======================================================================================

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
  , col = "#A52A2A" # Color curve in auburn
  , plot_p_limit = 1 - 0.9999
  , plot_counternull = FALSE
  , title = NULL
  , ylab = NULL
  , ylab_sec = NULL
  , inverted = FALSE
  , x_scale = "default"
  , plot = TRUE
)

#=======================================================================================
# 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,