Package ‘qqconf’

August 17, 2021

Type Package
Title Creates Simultaneous Testing Bands for QQ-Plots
Version 1.0.0
Description Provides functionality for creating Quantile-Quantile (QQ) and Probability-Probability (PP) plots with simultaneous testing bands to assess significance of sample deviation from a reference distribution.
License GPL-3
Depends R (>= 3.0.0)
Encoding UTF-8
RoxygenNote 7.1.1
Imports dplyr (>= 1.0.0), magrittr (>= 1.5), rlang (>= 0.4.9), ggplot2 (>= 3.0.0), MASS (>= 7.3-50), robustbase (>= 0.93-4)
Collate 'one_sided.R' 'ppplot.R' 'qqconf-package.R' 'qqplot.R' 'two_sided.R'

NeedsCompilation yes
Author Eric Weine [aut, cre],
Mary Sara McPeek [aut],
Abney Mark [aut]
Maintainer Eric Weine <ericweine15@gmail.com>
Repository CRAN
Date/Publication 2021-08-17 07:20:08 UTC

R topics documented:

check Bounds_one_sided .............................................. 2
check Bounds_two_sided ............................................. 2
get asymptotic approx corrected_alpha ............................. 3
get Bounds_one_sided ................................................. 3
get Bounds_two_sided ................................................ 4
get level from Bounds_one_sided .................................... 6
get level from Bounds_two_sided .................................... 6
check_bounds_one_sided

Check Validity of One-Sided Bounds

Description
Given bounds for a one sided test, this checks that none of the bounds fall outside of [0, 1].

Usage
check_bounds_one_sided(upper_bounds)

Arguments
- upper_bounds: Numeric vector where the ith component is the upper bound for the ith order statistic.

Value
None

check_bounds_two_sided

Check Validity of Two-Sided Bounds

Description
Given bounds for a two sided test, this checks that none of the bounds fall outside of [0, 1] and that all upper bounds are greater than the corresponding lower bounds. This also ensures the the length of the bounds are the same. This not meant to be called by the user.

Usage
check_bounds_two_sided(lower_bounds, upper_bounds)

Arguments
- lower_bounds: Numeric vector where the ith component is the lower bound for the ith order statistic.
- upper_bounds: Numeric vector where the ith component is the lower bound for the ith order statistic.
get_asymptotic_approx_corrected_alpha

Calculates Approximate Local Level

Description

This function uses the approximation from Gontscharuk & Finner’s Asymptotics of goodness-of-fit tests based on minimum p-value statistics (2017) to approximate local levels for finite sample size. We use these authors constants for $\alpha = .1$, and .05, and for $\alpha = .01$ we use a slightly different approximation.

Usage

get_asymptotic_approx_corrected_alpha(n, alpha)

Arguments

- **n**: Number of tests to do
- **alpha**: Global type I error rate $\alpha$ of the tests

Value

Approximate local level

get_bounds_one_sided

Calculates Rejection Region of One-Sided Equal Local Levels Test

Description

The context is that $n$ i.i.d. observations are assumed to be drawn from some distribution on the unit interval with c.d.f. $F(x)$, and it is desired to test the null hypothesis that $F(x) = x$ for all $x$ in $(0,1)$, referred to as the "global null hypothesis," against the alternative $F(x) > x$ for at least one $x$ in $(0, 1)$. An "equal local levels" test is used, in which each of the $n$ order statistics is tested for significant deviation from its null distribution by a one-sided test with significance level $\eta$. The global null hypothesis is rejected if at least one of the order statistic tests is rejected at level $\eta$, where $\eta$ is chosen so that the significance level of the global test is $\alpha$. Given the size of the dataset $n$ and the desired global significance level $\alpha$, this function calculates the local level $\eta$ and the acceptance/rejection regions for the test. The result is a set of lower bounds, one for each order statistic. If at least one order statistic falls below the corresponding bound, the global test is rejected. Note that the code may be slow for $n > 500$. 
get_bounds_two_sided

Calculates Rejection Region of Two-Sided Equal Local Levels Test.

Description

The context is that \( n \) i.i.d. observations are assumed to be drawn from some distribution on the unit interval with c.d.f. \( F(x) \), and it is desired to test the null hypothesis that \( F(x) = x \) for all \( x \) in \((0,1)\), referred to as the "global null hypothesis," against a two-sided alternative. An "equal local levels" test is used, in which each of the \( n \) order statistics is tested for significant deviation from its null distribution by a 2-sided test with significance level \( \eta \). The global null hypothesis is rejected if at least one of the order statistic tests is rejected at level \( \eta \), where \( \eta \) is chosen so that the significance level of the global test is \( \alpha \). Given the size of the dataset \( n \) and the desired global significance level \( \alpha \), this function calculates the local level \( \eta \) and the acceptance/rejection regions for the test. There are a set of \( n \) intervals, one for each order statistic. If at least one order statistic falls outside the corresponding interval, the global test is rejected.
Usage

get_bounds_two_sided(
  alpha,
  n,
  tol = 1e-08,
  max_it = 100,
  method = c("best_available", "approximate", "search")
)

Arguments

alpha | Desired global significance level of the test.

n | Size of the dataset.

tol | (Optional) Relative tolerance of the alpha level of the simultaneous test. Defaults to 1e-8.

max_it | (Optional) Maximum number of iterations of Binary Search Algorithm used to find the bounds. Defaults to 100 which should be much larger than necessary for a reasonable tolerance.

method | (Optional) Parameter indicating if the calculation should be done using a highly accurate approximation, "approximate", or if the calculations should be done using an exact binary search calculation, "search". The default is "best_available" (recommended), which uses the exact search when either (i) the approximation isn’t available or (ii) the approximation is available but isn’t highly accurate and the search method isn’t prohibitively slow (occurs for small to moderate n with alpha = .1). Of note, the approximate method is only available for alpha values of .1, .05, and .01. In the case of alpha = .05 or .01, the approximation is highly accurate for all values of n up to at least 10^6.

Value

A list with components

- lower_bound - Numeric vector of length n containing the lower bounds for the acceptance regions of the test of each order statistic.
- upper_bound - Numeric vector of length n containing the upper bounds for the acceptance regions of the test of each order statistic.
- x - Numeric vector of length n containing the expectation of each order statistic. These are the x-coordinates for the bounds if used in a qq-plot. The value is c(1:n) / (n + 1).
- local_level - Significance level \( \eta \) of the local test on each individual order statistic. It is equal for all order statistics and will be less than alpha for all \( n > 1 \).

Examples

g...
get_level_from_bounds_one_sided

Calculates Global Significance Level From Simultaneous One-Sided Bounds for Rejection Region

Description

For a one-sided test of uniformity of i.i.d. observations on the unit interval, this function will determine the significance level as a function of the rejection region. Suppose $n$ observations are drawn i.i.d. from some CDF $F(x)$ on the unit interval, and it is desired to test the null hypothesis that $F(x) = x$ for all $x$ in $(0, 1)$ against the one-sided alternative $F(x) > x$. Suppose the acceptance region for the test is described by a set of lower bounds, one for each order statistic. Given the lower bounds, this function calculates the significance level of the test where the null hypothesis is rejected if at least one of the order statistics falls below its corresponding lower bound.

Usage

get_level_from_bounds_one_sided(bounds)

Arguments

- bounds: Numeric vector where the $i$th component is the lower bound for the $i$th order statistic. The components must be distinct values in $(0, 1)$ that are in ascending order.

Value

Global significance level

Examples

# For $X_1, X_2, X_3$ i.i.d. unif(0, 1),
# calculate $1 - P(X(1) > .1$ and $X(2) > .5$ and $X(3) > .8)$,
# where $X(1), X(2)$, and $X(3)$ are the order statistics.
get_level_from_bounds_one_sided(bounds = c(.1, .5, .8))

get_level_from_bounds_two_sided

Calculates Global Significance Level From Simultaneous Two-Sided Bounds for Rejection Region
Description
For a test of uniformity of i.i.d. observations on the unit interval, this function will determine the
significance level as a function of the rejection region. Suppose \( n \) observations are drawn i.i.d. from
some CDF \( F(x) \) on the unit interval, and it is desired to test the null hypothesis that \( F(x) = x \) for all \( x \) in \((0, 1)\) against a two-sided alternative. Suppose the acceptance region for the test is described
by a set of intervals, one for each order statistic. Given the bounds for these intervals, this function
calculates the significance level of the test where the null hypothesis is rejected if at least one of the
order statistics is outside its corresponding interval.

Usage

get_level_from_bounds_two_sided(lower_bounds, upper_bounds, is_ell = FALSE)

Arguments

- **lower_bounds**: Numeric vector where the \( i \)th component is the lower bound for the acceptance
  interval for the \( i \)th order statistic. The components must be distinct values in \((0, 1)\)
  that are in ascending order.

- **upper_bounds**: Numeric vector of the same length as `lower_bounds` where the \( i \)th component
  is the upper bound for the acceptance interval for the \( i \)th order statistic. The
  values must be in ascending order, the \( i \)th component must be greater than the
  \( i \)th component of the `lower_bounds` vector and less than 1, and the elements of
  `c(lower_bounds, upper_bounds)` must all be distinct.

- **is_ell**: (Optional) Boolean parameter indicating whether the bounds were derived as a
  result of conducting an \( \eta \) level two-sided symmetric test on each order statistic
  (where \( \eta \) is the same for each order statistic). If the parameter is set to TRUE, a
  speedup will be used that cuts the computation time roughly in half. However,
  this will return an incorrect answer if set to TRUE and bounds are input that
  are not derived from an equal local levels test. The actual condition needed for
  `get_level_from_bounds_two_sided` to return a correct answer with `is_ell`
  set to TRUE is `upper_bounds == 1 - rev(lower_bounds)`.

Value

Global Significance Level \( \alpha \)

Examples

# For X1, X2 iid unif(0,1), calculate 1 - P(.1 < min(X1, X2) < .6 and .5 < max(X1, X2) < .9)
get_level_from_bounds_two_sided(lower_bounds = c(.1, .5), upper_bounds = c(.6, .9))

# Finds the global significance level corresponding to the local level \eta.  
# Suppose we reject the null hypothesis that X1, ..., Xn are iid unif(0, 1) if and only if at least
# one of the order statistics X(i) is significantly different from
# its null distribution based on a level-\eta
# two-sided test, i.e. we reject if and only if X(i) is outside the interval
# (qbeta(\eta/2, i, n - i + 1), qbeta(1 - \eta/2, i, n - i + 1)) for at least one i.
# The lines of code below calculate the global significance level of
# the test (which is necessarily larger than \eta if n > 1).
n <- 100
teta <- .05
lb <- qbeta(teta / 2, c(1:n), c(n:1))
ub <- qbeta(1 - teta / 2, c(1:n), c(n:1))
get_level_from_bounds_two_sided(lower_bounds = lb, upper_bounds = ub, is_ell=TRUE)

---

monte_carlo_two_sided  
Monte Carlo Simulation for Two-Sided Test

**Description**

Given bounds for a two sided test on uniform order statistics, this computes the Type I Error Rate \( \alpha \) using simulations.

**Usage**

```r
monte_carlo_two_sided(lower_bounds, upper_bounds, num_sims = 1e+06)
```

**Arguments**

- `lower_bounds`: Numeric vector where the ith component is the lower bound for the ith order statistic. The components must be distinct values in \((0, 1)\) that are in ascending order.
- `upper_bounds`: Numeric vector where the ith component is the lower bound for the ith order statistic. The values must be in ascending order and the ith component must be larger than the ith component of the lower bounds.
- `num_sims`: (Optional) Number of simulations to be run, 1 Million by default.

**Value**

Type I Error Rate \( \alpha \)

---

pp_conf_plot  
PP Plot with Simultaneous and Pointwise Testing Bounds.

**Description**

Create a pp-plot with with a shaded simultaneous acceptance region and, optionally, lines for a point-wise region. The observed values are plotted against their expected values had they come from the specified distribution.
**Usage**

```r
pp_conf_plot(
  obs,
  distribution = pnorm,
  method = c("ell", "ks"),
  alpha = 0.05,
  difference = FALSE,
  log10 = FALSE,
  right_tail = FALSE,
  add = FALSE,
  dparams = list(),
  bounds_params = list(),
  line_params = list(),
  plot_pointwise = FALSE,
  pointwise_lines_params = list(),
  points_params = list(),
  polygon_params = list(border = NA, col = "gray"),
  ...
)
```

**Arguments**

- **obs**: The observed data.
- **distribution**: The probability function for the specified distribution. Defaults to `pnorm`. Custom distributions are allowed as long as all parameters are supplied in `dparams`.
- **method**: Method for simultaneous testing bands. Must be either "ell" (equal local levels test), which applies a level $\eta$ pointwise test to each order statistic such that the Type I error of the global test is $\alpha$, or "ks" to apply a Kolmogorov-Smirnov test. For $\alpha = .01$, .05, and .1, "ell" is recommended.
- **alpha**: Type I error of global test of whether the data come from the reference distribution.
- **difference**: Whether to plot the difference between the observed and expected values on the vertical axis.
- **log10**: Whether to plot axes on -log10 scale (e.g. to see small p-values).
- **right_tail**: This parameter is only used if `log10` is TRUE. When TRUE, the x-axis is -log10(1 - Expected Probability) and the y-axis is -log10(1 - Observed Probability). When FALSE (default) the x-axis is -log10(Expected Probability) and the y-axis is -log10(Observed Probability). The parameter should be set to TRUE to make observations in the right tail of the distribution easier to see, and set to false to make the observations in the left tail of the distribution easier to see.
- **add**: Whether to add points to an existing plot.
- **dparams**: List of additional parameters for the probability function of the distribution (e.g. `df=1`). Note that if any parameters of the distribution are specified, parameter estimation will not be performed on the unspecified parameters, and instead they will take on the default values set by the distribution function. For the uniform distribution, parameter estimation is not performed, and the default parameters
are max = 1 and min = 0. For other distributions parameters will be estimated if
not provided. For the normal distribution, we estimate the mean as the median
and the standard deviation as $\hat{S}_n$ from the paper by Rousseeuw and Croux 1993
"Alternatives to the Median Absolute Deviation". For all other distributions be-
sides uniform and normal, the code uses MLE to estimate the parameters. Note
that estimation is not implemented for custom distributions, so all parameters of
the distribution must be provided by the user.

- **bounds_params**: List of optional parameters for get_bounds_two_sided (i.e. tol, max_it, method).
- **line_params**: Parameters passed to the line function to modify the line that indicates a perfect
  fit of the reference distribution.
- **plot_pointwise**: Boolean indicating whether pointwise bounds should be added to the plot
- **pointwise_lines_params**: Parameters passed to the lines function that modifies pointwise bounds when
  plot_pointwise is set to TRUE.
- **points_params**: Parameters to be passed to the points function to plot the data.
- **polygon_params**: Parameters to be passed to the polygon function to construct simultaneous con-
  fidence region. By default border is set to NA and col is set to grey.
- **Additional parameters passed to the plot function.

### Details

If any of the points of the pp-plot fall outside the simultaneous acceptance region for the selected
level alpha test, that means that we can reject the null hypothesis that the data are i.i.d. draws
from the specified distribution. If difference is set to TRUE, the vertical axis plots the observed
probability minus expected probability. If pointwise bounds are used, then on average, alpha * n
of the points will fall outside the bounds under the null hypothesis, so the chance that the pp-plot
has any points falling outside of the pointwise bounds is typically much higher than alpha under the
null hypothesis. For this reason, a simultaneous region is preferred.

### Value

None, PP plot is produced.

### Examples

```r
set.seed(0)
smp <- rnorm(100)

# Plot PP plot against normal distribution with mean and variance estimated
pp_conf_plot(
  obs=smp,
  distribution = pnorm
)

# Make same plot on -log10 scale to highlight the left tail,
# with radius of plot circles also reduced by .5
pp_conf_plot(
  obs=smp,
```
Description

Create a qq-plot with with a shaded simultaneous acceptance region and, optionally, lines for a point-wise region. The observed values are plotted against their expected values had they come from the specified distribution.

Usage

```r
qq_conf_plot(
  obs,
  distribution = qnorm,
  method = c("ell", "ks"),
  alpha = 0.05,
  difference = FALSE,
  log10 = FALSE,
  right_tail = FALSE,
  add = FALSE,
  dparams = list(),
  bounds_params = list(),
  line_params = list(),
  plot_pointwise = FALSE,
  pointwise_lines_params = list()
)
```
points_params = list(),
polygon_params = list(border = NA, col = "gray"),
...  
)

Arguments

obs
The observed data.
distribution
The quantile function for the specified distribution. Defaults to qnorm. Custom distributions are allowed as long as all parameters are supplied in dparams.
method
Method for simultaneous testing bands. Must be either "ell" (equal local levels test), which applies a level \( \eta \) pointwise test to each order statistic such that the Type I error of the global test is \( \alpha \), or "ks" to apply a Kolmogorov-Smirnov test. For \( \alpha = .01, .05, \) and \( .1 \), "ell" is recommended.
alpha
Type I error of global test of whether the data come from the reference distribution.
difference
Whether to plot the difference between the observed and expected values on the vertical axis.
log10
Whether to plot axes on -log10 scale (e.g. to see small p-values). Can only be used for strictly positive distributions.
right_tail
This parameter is only used if log10 is TRUE. When TRUE, the x-axis is -log10(1 - Expected Quantile) and the y-axis is -log10(1 - Observed Quantile). When FALSE (default) the x-axis is -log10(Expected Quantile) and the y-axis is -log10(Observed Quantile). The parameter should be set to TRUE to make observations in the right tail of the distribution easier to see, and set to false to make the observations in the left tail of the distribution easier to see.
add
Whether to add points to an existing plot.
dparams
List of additional parameters for the quantile function of the distribution (e.g. df=1). Note that if any parameters of the distribution are specified, parameter estimation will not be performed on the unspecified parameters, and instead they will take on the default values set by the distribution function. For the uniform distribution, parameter estimation is not performed, and the default parameters are \( \max = 1 \) and \( \min = 0 \). For other distributions parameters will be estimated if not provided. For the normal distribution, we estimate the mean as the median and the standard deviation as \( S_n \) from the paper by Rousseeuw and Croux 1993 "Alternatives to the Median Absolute Deviation". For all other distributions besides uniform and normal, the code uses MLE to estimate the parameters. Note that estimation is not implemented for custom distributions, so all parameters of the distribution must be provided by the user.
bounds_params
List of optional parameters for get_bounds_two_sided (i.e. tol, max_it, method).
line_params
Parameters passed to the lines function to modify the line that indicates a perfect fit of the reference distribution.
plot_pointwise
Boolean indicating whether pointwise bounds should be added to the plot
pointwise_lines_params
Parameters passed to the lines function that modifies pointwise bounds when plot_pointwise is set to TRUE.
points_params  Parameters to be passed to the points function to plot the data.

polygon_params  Parameters to be passed to the polygon function to construct simultaneous confidence region. By default border is set to NA and col is set to grey.

Additional parameters passed to the plot function.

Details

If any of the points of the qq-plot fall outside the simultaneous acceptance region for the selected level alpha test, that means that we can reject the null hypothesis that the data are i.i.d. draws from the specified distribution. If difference is set to TRUE, the vertical axis plots the observed quantile minus expected quantile. If pointwise bounds are used, then on average, alpha * n of the points will fall outside the bounds under the null hypothesis, so the chance that the qq-plot has any points falling outside of the pointwise bounds is typically much higher than alpha under the null hypothesis. For this reason, a simultaneous region is preferred.

Value

None, QQ plot is produced.

Examples

```r
set.seed(0)
smp <- runif(100)

# Plot QQ plot against uniform(0, 1) distribution
qq_conf_plot(
  obs=smp,
  distribution = qunif
)

# Make same plot on -log10 scale to highlight small p-values, # with radius of plot circles also reduced by .5
qq_conf_plot(
  obs=smp,
  distribution = qunif,
  points_params = list(cex = .5),
  log10 = TRUE
)

# Make same plot with difference between observed and expected values on the y-axis
qq_conf_plot(
  obs=smp,
  distribution = qunif,
  difference = TRUE
)

# Make same plot with sample plotted as a blue line, expected value line plotted as a red line, # and with pointwise bounds plotted as black lines
qq_conf_plot(
  obs=smp,
  distribution = qunif,
)```
plot_pointwise = TRUE,
method = "ell",
points_params = list(col="blue", type="l"),
line_params = list(col="red")
)
Index

check_bounds_one_sided, 2
check_bounds_two_sided, 2

get_asymptotic_approx_corrected_alpha, 3
get_bounds_one_sided, 3
get_bounds_two_sided, 4
get_level_from_bounds_one_sided, 6
get_level_from_bounds_two_sided, 6

monte_carlo_two_sided, 8

pp_conf_plot, 8

qq_conf_plot, 11