# Package ‘qqconf’

April 15, 2023

**Type** Package

**Title** Creates Simultaneous Testing Bands for QQ-Plots

**Version** 1.3.2

**Description** Provides functionality for creating Quantile-Quantile (QQ) and Probability-Probability (PP) plots with simultaneous
testing bands to asses significance of sample deviation from a reference distribution <doi:10.18637/jss.v106.i10>.

**License** GPL-3

**Depends** R (>= 4.0.0)

**SystemRequirements** fftw3 (>= 3.1.2)

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Imports** MASS (>= 7.3-50), robustbase (>= 0.93-4), Rcpp

**Suggests** knitr, rmarkdown, distr (>= 2.8.0)

**Collate** 'one_sided.R' 'ppplot.R' 'qqconf-package.R' 'qqplot.R' 'RcppExports.R' 'two_sided.R' 'utils.R'

**VignetteBuilder** knitr

**LinkingTo** Rcpp

**URL** `https://github.com/eweine/qqconf`

**BugReports** `https://github.com/eweine/qqconf/issues`

**NeedsCompilation** yes

**Author** Eric Weine [aut, cre],
Mary Sara McPeek [aut],
Abney Mark [aut]

**Maintainer** Eric Weine <`ericweine15@gmail.com`>

**Repository** CRAN

**Date/Publication** 2023-04-14 23:00:21 UTC

# R topics documented:

between . . . . . . 2
check_bounds_one_sided . . . . . . 3
check_bounds_two_sided . . . . . . 3
estimate_params_from_data . . . . . . 4
get_asymptotic_approx_corrected_alpha . . . . . . 4
get_best_available_prob_pts_method . . . . . . 5
get_bounds_one_sided . . . . . . 5
get_bounds_two_sided . . . . . . 6
get_extended_quantile . . . . . . 8
get_level_from_bounds_one_sided . . . . . . 8
get_level_from_bounds_two_sided . . . . . . 9
get_mass_name_from_distr . . . . . . 10
get_qq_band . . . . . . 11
get_qq_distribution_from_pp_distribution . . . . . . 13
monte_carlo_two_sided . . . . . . 14
pp_conf_plot . . . . . . 14
qq_conf_plot . . . . . . 17

**Index** **21**

---

## `between` *Shorthand for two numerical comparisons*

---

### Description

Shorthand for two numerical comparisons

### Usage

```
between(x, gte, lte)
```

### Arguments

| | |
|---|---|
| `x` | numeric value |
| `gte` | lower bound |
| `lte` | upper bound |

### Value

boolean

---

`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. |

## Value

None

## estimate_params_from_data
*Estimate Parameters from Data*

### Description

For select distributions, parameters are estimated from data. Generally, the MLEs are used. However, for the normal distribution we use robust estimators.

### Usage

```
estimate_params_from_data(distr_name, obs)
```

### Arguments

| | |
|---|---|
| distr_name | MASS name of distribution |
| obs | observation vector |

### Value

list of distribution parameters

## 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_best_available_prob_pts_method` *Get Best Available Method for Probability Points*

---

## Description

Determines name of best method for obtaining expected points for a qq or pp plot.

## Usage

```
get_best_available_prob_pts_method(dist_name)
```

## Arguments

| | |
|---|---|
| `dist_name` | character name of distribution |

## Value

character name of best expected points method

---

`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.

## Usage

```
get_bounds_one_sided(alpha, n, tol = 1e-08, max_it = 100)
```

### 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. |

### Value

A list with components

- bound - Numeric vector of length `n` containing the lower bounds of the acceptance regions for 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

```
get_bounds_one_sided(alpha = .05, n = 10, max_it = 50)
```

---

## `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. Used only if `method` is set to "search" or if method is set to "best_available" and the best available method is a search. |
| `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. Used only if `method` is set to "search" or if method is set to "best_available" and the best available method is a search. |
| `method` | (optional) Argument 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 pp-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

```
get_bounds_two_sided(alpha = .05, n = 100)
```

## get_extended_quantile *Get Quantile for First and Last Point of QQ or PP Plot*

### Description

Get Quantile for First and Last Point of QQ or PP Plot

### Usage

```
get_extended_quantile(exp_pts_method, n)
```

### Arguments

| | |
|---|---|
| exp_pts_method | method used to derive expected points |
| n | sample size |

### Value

list with low and high point

## 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 ith component is the lower bound for the ith order statistic. The components must lie in [0, 1], and each component must be greater than or equal to the previous one. |

## Details

Uses the method of Moscovich and Nadler (2016) as implemented in Crossprob (Moscovich 2020).

## Value

Global significance level

## References


- Moscovich, Amit, and Boaz Nadler. "Fast calculation of boundary crossing probabilities for Poisson processes." Statistics & Probability Letters 123 (2017): 177-182.
- Amit Moscovich (2020). Fast calculation of p-values for one-sided Kolmogorov-Smirnov type statistics. arXiv:2009.04954


## Examples

```
# For X1, X2, X3 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)
```

## Arguments

| | |
|---|---|
| `lower_bounds` | Numeric vector where the ith component is the lower bound for the acceptance interval for the ith order statistic. The components must lie in [0, 1], and each component must be greater than or equal to the previous one. |

upper_bounds — Numeric vector of the same length as lower_bounds where the ith component is the upper bound for the acceptance interval for the ith order statistic. The components must lie in [0, 1], and each component must be greater than or equal to the previous one. In addition, the ith component of upper_bounds must be greater than or equal to the ith component of lower_bounds.

## Details

Uses the method of Moscovich and Nadler (2016) as implemented in Crossprob (Moscovich 2020).

## Value

Global Significance Level $\alpha$

## References


- Moscovich, Amit, and Boaz Nadler. "Fast calculation of boundary crossing probabilities for Poisson processes." Statistics & Probability Letters 123 (2017): 177-182.
- Amit Moscovich (2020). Fast calculation of p-values for one-sided Kolmogorov-Smirnov type statistics. arXiv:2009.04954


## 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
eta <- .05
lb <- qbeta(eta / 2, c(1:n), c(n:1))
ub <- qbeta(1 - eta / 2, c(1:n), c(n:1))
get_level_from_bounds_two_sided(lower_bounds = lb, upper_bounds = ub)
```

---

# get_mass_name_from_distr — *Convert R Distribution Function to MASS Distribution Name*

---

## Description

Convert R Distribution Function to MASS Distribution Name

### Usage

```
get_mass_name_from_distr(distr, band_type)
```

### Arguments

| | |
|---|---|
| distr | R distribution function (e.g. qnorm or pnorm) |
| band_type | one of "qq" (for quantile functions) or "pp" ( for probability functions). |

### Value

string of MASS distribution name

---

## get_qq_band *Create QQ Plot Testing Band*

---

### Description

Flexible interface for creating a testing band for a Quantile-Quantile (QQ) plot.

### Usage

```
get_qq_band(
  n,
  obs,
  alpha = 0.05,
  distribution = qnorm,
  dparams = list(),
  ell_params = list(),
  band_method = c("ell", "ks", "pointwise"),
  prob_pts_method = c("best_available", "normal", "uniform", "median")
)
```

### Arguments

| | |
|---|---|
| n, obs | either a number of observations (specified by setting n), or a numeric vector of observations (specified by setting obs). One argument must be specified. If all parameters of distribution are known, then the testing band only depends on the number of observations n. Thus, providing n is simpler when all parameters of distribution are known and specified via dparams (or when using all default parameter choices of distribution is desired). If estimating parameters from the data is preferred, obs should be specified and estimation will take place as described in the documentation for argument dparams. |
| alpha | (optional) desired significance level of the testing band. If method is set to "ell" or "ks", then this is the global significance level of the testing band. If method is set to "pointwise", then the band is equivalent to simply conducting a level alpha test on each order statistic individually. Pointwise bands will generally have much larger global Type I error than alpha. Defaults to .05. |

distribution
: The quantile function for the specified distribution. Defaults to qnorm, which is appropriate for testing normality of the observations in a QQ plot.

dparams
: (optional) List of additional arguments for the distribution function (e.g. df=1). If obs is not specified and this argument is left blank, the default arguments of distribution are used. If obs is specified and this argument is left blank, parameters are estimated from the data (except if distribution is set to qunif, in which case no estimation occurs and the default parameters are max = 1 and min = 0). For the normal distribution, we estimate the mean as the median and the standard deviation as $Sn$ 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 if any parameters of the distribution are specified in dparams, parameter estimation will not be performed on the unspecified parameters, and instead they will take on the default values set by distribution.

ell_params
: (optional) list of optional arguments for get_bounds_two_sided (i.e. tol, max_it, method). Only used if method is set to "ell"

band_method
: (optional) method for creating the testing band. The default, "ell" uses the equal local levels method (see get_bounds_two_sided for more information). "ks" uses the Kolmogorov-Smirnov test. "pointwise" uses a pointwise band (see documentation for argument alpha for more information). "ell" is recommended and is the default.

prob_pts_method
: (optional) method used to get probability points for use in a QQ plot. The quantile function will be applied to these points to get the expected values. When this argument is set to "normal" (recommended for a normal QQ plot) ppoints(n) will be used, which is what most other plotting software uses. When this argument is set to "uniform" (recommended for a uniform QQ plot) ppoints(n, a=0), which are the expected values of the order statistics of Uniform(0, 1), will be used. Finally, when this argument is set to "median" (recommended for all other distributions) qbeta(.5, c(1:n), c(n:1)) will be used. Under the default setting, "best_available", the probability points as recommended above will be used. Note that "median" is suitable for all distributions and is particularly recommended when alpha is large.

## Value

A list with components

- lower_bound - Numeric vector of length n containing the lower bounds for the acceptance regions of the test corresponding to each order statistic. These form the lower boundary of the testing band for the QQ-plot.
- upper_bound - Numeric vector of length n containing the upper bounds for the acceptance regions of the test corresponding to each order statistic. These form the upper boundary of the testing band for the QQ-plot.
- expected_value - Numeric vector of length n containing the exact or approximate expectation (or median) of each order statistic, depending on how prob_pts_method is set. These are the x-coordinates for both the bounds and the data points if used in a qq-plot. Note that if adding a band to an already existing plot, it is essential that the same x-coordinates be used for the

bounds as were used to plot the data. Thus, if some other x-coordinates have been used to plot the data those same x-coordinates should always be used instead of this vector to plot the bounds.

- dparams - List of arguments used to apply `distribution` to `obs` (if observations are provided). If the user provides parameters, then these parameters will simply be returned. If parameters are estimated from the data, then the estimated parameters will be returned.

## Examples

```
# Get ell level .05 QQ testing band for normal(0, 1) distribution with 100 observations
band <- get_qq_band(n = 100)

# Get ell level .05 QQ testing band for normal distribution with unknown parameters
obs <- rnorm(100)
band <- get_qq_band(obs = obs)

# Get ell level .05 QQ testing band for t(2) distribution with 100 observations
band <- get_qq_band(
  n = 100, distribution = qt, dparams = list(df = 2)
)
```

---

`get_qq_distribution_from_pp_distribution`
*Get Quantile Distribution from Probability Distribution*

---

## Description

Get Quantile Distribution from Probability Distribution

## Usage

```
get_qq_distribution_from_pp_distribution(dname)
```

## Arguments

| | |
|---|---|
| `dname` | probability distribution (e.g. `pnorm`) |

## Value

quantile distribution (e.g. `qnorm`).

## 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

```
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

```
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"),
  prob_pts_method = c("uniform", "median", "normal"),
  ...
)
```

## 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. "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 argument 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 argument 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 arguments 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 $Sn$ 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 arguments for get_bounds_two_sided. (i.e. tol, max_it, method). |

`line_params` arguments 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`
arguments passed to the `lines` function that modifies pointwise bounds when plot_pointwise is set to TRUE.

`points_params` arguments to be passed to the `points` function to plot the data.

`polygon_params` Arguments 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.

`prob_pts_method`
(optional) method used to get probability points for plotting. The default value, `"uniform"`, results in `ppoints(n, a=0)`, which are the expected values of the order statistics of Uniform(0, 1). When this argument is set to `"median"`, `qbeta(.5, c(1:n), c(n:1))`, the medians of the order statistics of Uniform(0, 1) will be used. For a PP plot, there is no particular theoretical justification for setting this argument to `"normal"`, which results in `ppoints(n)`, but it is an option because it is used in some other packages. When `alpha` is large, `"median"` is recommended.

`...` Additional arguments 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.

## References

Weine, E., McPeek, MS., & Abney, M. (2023). Application of Equal Local Levels to Improve Q-Q Plot Testing Bands with R Package qqconf Journal of Statistical Software, 106(10). https://doi:10.18637/jss.v106.i10

## Examples

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

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

```
# 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,
  log10 = TRUE,
  points_params = list(cex = .5)
)

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

# Make same plot with samples plotted as a blue line, expected value line plotted as a red line,
# and pointwise bounds plotted as black lines
pp_conf_plot(
  obs=smp,
  plot_pointwise = TRUE,
  points_params = list(col="blue", type="l"),
  line_params = list(col="red")
)
```

---

`qq_conf_plot` *QQ Plot with Simultaneous and Pointwise Testing Bounds.*

---

## 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

```
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"),
  prob_pts_method = c("best_available", "normal", "uniform", "median"),
  ...
)
```

**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. "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 argument 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 argument should be set to `TRUE` only when the support of the distribution lies in (0, 1), and one wants to make observations in the right tail of the distribution easier to see. The argument should be set to `FALSE` when one wants to make observations in the left tail of the distribution easier to see.

add
: Whether to add points to an existing plot.

dparams
: List of additional arguments 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 $Sn$ 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 arguments for `get_bounds_two_sided` (i.e. `tol`, `max_it`, `method`).

line_params
: Arguments 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`
: Arguments passed to the `lines` function that modifies pointwise bounds when plot_pointwise is set to TRUE.

`points_params`
: Arguments to be passed to the `points` function to plot the data.

`polygon_params`
: Arguments 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.

`prob_pts_method`
: (optional) method used to get probability points for plotting. The quantile function will be applied to these points to get the expected values. When this argument is set to `"normal"` (recommended for a normal QQ plot) `ppoints(n)` will be used, which is what most other plotting software uses. When this argument is set to `"uniform"` (recommended for a uniform QQ plot) `ppoints(n, a=0)`, which are the expected values of the order statistics of Uniform(0, 1), will be used. Finally, when this argument is set to `"median"` (recommended for all other distributions) `qbeta(.5, c(1:n), c(n:1))` will be used. Under the default setting, `"best_available"`, the probability points as recommended above will be used. Note that `"median"` is suitable for all distributions and is particularly recommended when alpha is large.

`...`
: Additional arguments 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.

## References

Weine, E., McPeek, MS., & Abney, M. (2023). Application of Equal Local Levels to Improve Q-Q Plot Testing Bands with R Package qqconf Journal of Statistical Software, 106(10). https://doi:10.18637/jss.v106.i10

## Examples

```
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,
  points_params = list(col="blue", type="l"),
  line_params = list(col="red")
)
```

# Index

between, 2

check_bounds_one_sided, 3
check_bounds_two_sided, 3

estimate_params_from_data, 4

get_asymptotic_approx_corrected_alpha, 4
get_best_available_prob_pts_method, 5
get_bounds_one_sided, 5
get_bounds_two_sided, 6
get_extended_quantile, 8
get_level_from_bounds_one_sided, 8
get_level_from_bounds_two_sided, 9
get_mass_name_from_distr, 10
get_qq_band, 11
get_qq_distribution_from_pp_distribution, 13

monte_carlo_two_sided, 14

pp_conf_plot, 14

qq_conf_plot, 17