Package: qqtest (via r-universe)

October 14, 2024

Type Package

Title Self Calibrating Quantile-Quantile Plots for Visual Testing

Version 1.2.0

Date 2020-03-14

Maintainer Wayne Oldford <rwoldford@uwaterloo.ca>

Description Provides the function qqtest which incorporates uncertainty in its qqplot display(s) so that the user might have a better sense of the evidence against the specified distributional hypothesis. qqtest draws a quantile quantile plot for visually assessing whether the data come from a test distribution that has been defined in one of many ways. The vertical axis plots the data quantiles, the horizontal those of a test distribution. The default behaviour generates 1000 samples from the test distribution and overlays the plot with shaded pointwise interval estimates for the ordered quantiles from the test distribution. A small number of independently generated exemplar quantile plots can also be overlaid. Both the interval estimates and the exemplars provide different comparative information to assess the evidence provided by the qqplot for or against the hypothesis that the data come from the test distribution (default is normal or gaussian).
Finally, a visual test of significance (a lineup plot) can also be displayed to test the null hypothesis that the data come from the test distribution.

LazyData true

License GPL-3

Depends R (>= 2.10.0)

Imports grDevices, stats

NeedsCompilation no

Author Wayne Oldford [aut, cre]

Encoding UTF-8

RoxygenNote 7.0.2

Suggests knitr, rmarkdown

VignetteBuilder knitr

URL https://github.com/rwoldford/qqtest,
https://rwoldford.github.io/qqtest/

Repository https://rwoldford.r-universe.dev

RemoteUrl https://github.com/rwoldford/qqtest

RemoteRef HEAD

RemoteSha f3737db73bfd00e36067d394d749a7232c3f3bb9

Contents

`bacteria` *Bacteria from Delaware River water entering the Torresdale Filter of the Philadelphia water supply 1913.*

Description

The number of bacteria per cubic centimetre was measured daily for river water entering the Torresdale filter of the Philadelphia water supply through the whole of 1913.

Usage

```
bacteria
```

Format

A data frame with 22 rows and 2 variates:

count Number of bacteria per cc.

percentTime Percent of days (out of 365) having bacteria count less than or equal to the measured count.

Details

Rather than the individual daily results, recorded here are 22 values together with the percentage of days whose value was less than or equal to the recorded value. These quantiles are therefore based on 365 daily measurements.

Values were taken from a logarithmic-normal probability plot dated January 21, 1915 as it appeared in Figure 22 of George C. Whipple's 1916 (Part 2) paper on the "Element of Chance in Sanitation". This paper introduces logarithimic-normal probability paper (or a log-normal qqplot).

`with(bacteria, plot(qnorm(percentTime/100), log(count,10), type="o"))` reproduces Whipple's 1915 plot.

`with(bacteria, qqtest(data=count, p=percentTime/100, np=365, dist="log-normal", type="o"))` will effect a qqtest plot for this data. More detail can be had from: `with(bacteria, qqtest(data=log(count, 10), p=percentTime/100, np=365, dist="normal", type="o"))`

Source

"The Element of Chance in Sanitation", George C. Whipple, Journal of the Franklin Institue, Volume 182, July and August (1916), pp. 37-59 and 205-227. Data taken directly from Figure 22, page 209.

`dkay` `dkay` *The density function of the K distribution*

Description

The K density function on `df` degrees of freedom and non-centrality parameter `ncp`.

A K distribution is the square root of a chi-square divided by its degrees of freedom. That is, if x is chi-squared on m degrees of freedom, then y = sqrt(x/m) is K on m degrees of freedom. Under standard normal theory, K is the distribution of the pivotal quantity s/sigma where s is the sample standard deviation and sigma is the standard deviation parameter of the normal density. K is the natural distribution for tests and confidence intervals about sigma. K densities are more nearly symmetric than are chi-squared and concentrate near 1. As the degrees of freedom increase, they become more symmetric, more concentrated, and more nearly normally distributed.

Usage

```
dkay(x, df, ncp = 0, log.p = FALSE)
```

Arguments

x	A vector of values at which to calculate the density.
df	Degrees of freedom (non-negative, but can be non-integer).
ncp	Non-centrality parameter (non-negative).
log.p	logical; if TRUE, probabilities are given as log(p).

Value

dkay gives the density evaluated at the values of x.

Invalid arguments will result in return value NaN, with a warning.

The length of the result is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments are recycled to the length of the result. Only the first elements of the logical arguments are used.

Note

All calls depend on analogous calls to chi-squared functions. See dchisq for details on non-centrality parameter calculations.

Examples

```
dkay(1, 20)
#
# See also the vignette on the "K-distribution"
#
```

---

hideLocation hideLocation *Obfuscating the true location of the real data.*

Description

Hides the true location of the data as a non-obvious calculation string.

Usage

```
hideLocation(trueLoc, nSubjects)
```

Arguments

trueLoc	A vector of one or more numbers in the set 1, 2, ..., nSubjects whose locations are to be hidden.
nSubjects	An integer larger than 1 used to define the set 1, 2, ..., nSubjects.

Value

Returns a character vector, each element being a string containing an obscure calculation which, if parsed and evaluated, would return the value of the corresponding number in `trueLoc`.

See Also

`revealLocation`

Examples

```
trueLoc <- hideLocation(3,100)
trueLoc
revealLocation(trueLoc)

n <- 200
trueLoc <- sample(1:n, 3)
trueLoc
ans <- hideLocation(trueLoc , n)
ans
revealLocation(ans)
```

`penicillin` *31 contrast sums from a 32 run 2^(5-0) factorial experiment on penicillin production.*

Description

Values are arranged in decreasing order of absolute magnitude. Name of the contrast effect is given as the row name of each value. Daniel(1959) uses the data to illustrate the use of half-normal plots. In his words: "We need, of course, some rule of inference that will help us to be objective in judging whether or not the largest effects are real."

Usage

```
penicillin
```

Format

A data frame with 31 rows and 1 variate:

value value of contrast for that row

Details

`qqtest(penicillin[1],dist="half-normal")` will effect Daniel's plot.

Source

"Use of Half-Normal Plots in Interpreting Factorial Two-Level Experiments", Cuthbert Daniel, Technometrics, Vol. 1, No. 4 (Nov., 1959), pp. 311-341. Daniel cites: Davies, O. L., Editor: Design and Analysis of Industrial Experiments, Second Edition, Oliver and Boyd, London, and Hafner, New York, 1956 as the original.

---

pkay pkay *The cumulative distribution function K distribution*

Description

The cumulative distribution function for the K distribution on `df` degrees of freedom having non-centrality parameter `ncp`.

A K distribution is the square root of a chi-square divided by its degrees of freedom. That is, if x is chi-squared on m degrees of freedom, then y = sqrt(x/m) is K on m degrees of freedom. Under standard normal theory, K is the distribution of the pivotal quantity s/sigma where s is the sample standard deviation and sigma is the standard deviation parameter of the normal density. K is the natural distribution for tests and confidence intervals about sigma. K densities are more nearly symmetric than are chi-squared and concentrate near 1. As the degrees of freedom increase, they become more symmetric, more concentrated, and more nearly normally distributed.

Usage

```
pkay(q, df, ncp = 0, upper.tail = FALSE, log.p = FALSE)
```

Arguments

`q`	A vector of quantiles at which to calculate the cumulative distribution.
`df`	Degrees of freedom (non-negative, but can be non-integer).
`ncp`	Non-centrality parameter (non-negative).
`upper.tail`	logical; if `TRUE`, instead of returning F(q) (the default), the upper tail probabilities 1-F(q) = Pr(Q>q) are returned.
`log.p`	logical; if `TRUE`, probabilities are given as log(p).

Value

`pkay` returns the value of the K cumulative distribution function, F(q), evaluated at `q` for the given `df` and `ncp`. If `upper.trail` = `TRUE` then the upper tail probabilities 1-F(q) = Pr(Q>q) are returned instead of F(q).

Invalid arguments will result in return value NaN, with a warning.

The length of the result is the maximum of the lengths of the numerical arguments.

The numerical arguments are recycled to the length of the result. Only the first elements of the logical arguments are used.

Note

All calls depend on analogous calls to chi-squared functions. See pchisq for details on non-centrality parameter calculations.

Examples

```
pkay(1, 20)
q <- seq(0.01, 1.8, 0.01)
#
# Plot the cdf for K(5)
u <- pkay(q,5)
plot(q, u, type="l",
     xlab="q", ylab="cumulative probability",
     xlim=range(q), ylim=c(0,1),
     main="K cdf")
#
# Add some other K cdfs
lines(q, pkay(q,10), lty=2)
lines(q, pkay(q,20), lty=3)
lines(q, pkay(q,30), lty=4)
legend("topleft",
       legend=c("df = 5", "df = 10", "df = 20", "df = 30"),
       lty=c(1,2,3,4),
       title="degrees of freedom",
       cex=0.75, bty="n")

#
# See the vignette for more on the "K-distribution"
#
```

---

primer *Automobile primer paint thickness quality control measurements.*

Description

Contains process control measurements of thickness of primer applied to automotive body parts in an auto factory. Twice daily, a set of 10 consecutive parts were selected and the thickness in mils (thousandths of an inch) were measured. For each set of 10 parts, the average (xbar) and the sample standard deviation (s) were also calculated and recorded. These summaries would be plotted in xbar or s control charts with suitably determined upper and lower control limits. Alternatively, for checking outliers a qqplot (via qqtest) could be used for either xbar or s.

Usage

```
primer
```

Format

A data frame with 20 rows and 14 variates:

day Day on which the parts were taken and measured.

batch Either the first or second set of 10 consecutive parts taken.

sel1 Thickness of primer in mils on the first part sampled in the specified batch of that day.

sel2 Thickness of primer in mils on the second part sampled in the specified batch of that day.

sel3 Thickness of primer in mils on the third part sampled in the specified batch of that day.

sel4 Thickness of primer in mils on the fourth part sampled in the specified batch of that day.

sel5 Thickness of primer in mils on the fifth part sampled in the specified batch of that day.

sel6 Thickness of primer in mils on the sixth part sampled in the specified batch of that day.

sel7 Thickness of primer in mils on the seventh part sampled in the specified batch of that day.

sel8 Thickness of primer in mils on the eighth part sampled in the specified batch of that day.

sel9 Thickness of primer in mils on the ninth part sampled in the specified batch of that day.

sel10 Thickness of primer in mils on the tenth part sampled in the specified batch of that day.

xbar Arithmetic average of the measurements of primer thickness of the 10 parts selected in the specified batch of that day.

s Sample standard deviation of the measurements of primer thickness of the 10 parts selected in the specified batch of that day.

Details

`with(primer,qqtest(xbar, main="Averages"))` will effect this plot for xbar. `with(primer,qqtest(s,dist="kay", df=9, main ="Standard deviations"))` will effect this plot for s.

Source

"Statistical Process Control - SPC", Automotive Industry Action Group(AIAG), Southfield MI, (1995), page 64.

---

pullstrength *Strength of pull for 519 males aged 23-26.*

Description

From measurements made by Francis Galton at the International Health Exhibition in 1884.

Usage

```
pullstrength
```

Format

A data frame with 7 rows and 4 variates:

strength Pull strength lower bound in pounds.

nCases Number of cases observed with pull strength between this bound and the next.

percentCases Percent of cases observed with pull strength between this bound and the next.

percentCumulative Cumulative percent of cases observed with pull strength up to this bound.

percentAdjustedCumulative Adjust Galton's cumulative percent to include only half the cases between this bound and the next.

Details

`qqtest(pullstrength$strength, p=pullstrength$percentCumulative/100, np=519, dist="uniform", main="Galton's ogive of pull strength for 519 males aged 23-26", xlab="Cumulative Proportions (Adjusted)",yAxisAsProbs=FALSE, ylab="Strength in lbs.", type="o")` will effect Galton's Ogive.

`qqtest(pullstrength$strength, p=pullstrength$percentAdjustedCumulative/100, np=519, dist="normal", main="Gaussian qqplot of pull strength for 519 males aged 23-26", xlab="Cumulative Proportions (Adjusted)",yAxisAsProbs=FALSE, ylab="Strength in lbs.", type="o")` will effect a normal qqplot for this data.

Source

"Natural Inheritance", Francis Galton, (1889), Table 1, page 199.

---

qkay `qkay` *The K distribution quantile function*

Description

Quantile function for the K distribution on `df` degrees of freedom having non-centrality parameter `ncp`.

A K distribution is the square root of a chi-square divided by its degrees of freedom. That is, if x is chi-squared on m degrees of freedom, then y = sqrt(x/m) is K on m degrees of freedom. Under standard normal theory, K is the distribution of the pivotal quantity s/sigma where s is the sample standard deviation and sigma is the standard deviation parameter of the normal density. K is the natural distribution for tests and confidence intervals about sigma. K densities are more nearly symmetric than are chi-squared and concentrate near 1. As the degrees of freedom increase, they become more symmetric, more concentrated, and more nearly normally distributed.

Usage

```
qkay(p, df, ncp = 0, upper.tail = FALSE, log.p = FALSE)
```

Arguments

p	A vector of probabilities at which to calculate the quantiles.
df	Degrees of freedom (non-negative, but can be non-integer).
ncp	Non-centrality parameter (non-negative).
upper.tail	logical; if TRUE, instead of returning F(x) (the default), the upper tail probabilities 1-F(x) = Pr(X>x) are returned.
log.p	logical; if TRUE, probabilities are given as log(p).

Value

qkay returns the quantiles at probabilities p for a K on df degrees of freedom and non-centrality parameter ncp.

Invalid arguments will result in return value NaN, with a warning.

The length of the result is the maximum of the lengths of the numerical arguments.

The numerical arguments are recycled to the length of the result. Only the first elements of the logical arguments are used.

Note

All calls depend on analogous calls to chi-squared functions. See qchisq for details on non-centrality parameter calculations.

Examples

```
p <- ppoints(30)
# Get the quantiles for these points
q5 <- qkay(p, 5)
plot(p, q5, main="Quantile plot of K(20)", ylim=c(0,max(q5)))
# Add quantiles from another K
points(p, qkay(p, 20), pch=19)

#
# Do these EXACT quantiles from a K(5) look like they might
# have been generated from K(20)?
qqtest(q5, dist="kay",df=20)

# How about compared to normal?
qqnorm(q5)
qqtest(q5)
# for this many degrees of freedom it looks a lot like
# a gaussian (normal) distribution

# And should look really good compared to the true distribution
qqtest(q5, dist="kay", df=5)
#
#
# But not so much like it came from a K on 1 degree of freedom
qqtest(q5, dist="kay",df=1)
```

```
#
# See the vignette for more on the "K-distribution"
#
```

---

qqtest — qqtest *A self-calibrated quantile-quantile plot for assessing distributional shape.*

Description

Draws a quantile-quantile plot for visually assessing whether the data come from a test distribution that has been defined in one of many ways.

The vertical axis plots the data quantiles, the horizontal those of a test distribution.

Interval estimates and exemplars provide different comparative information to assess the evidence provided by the qqplot against the hypothesis that the data come from the test distribution (default is normal or gaussian). Interval estimates provide test information related to individual quantiles, exemplars provide test information related to the shape of the quantile quantile curve.

Optionally, a visual test of significance (a lineup plot) can be displayed to provide a coarse level of significance for testing the null hypothesis that the data come from the test distribution.

The default behaviour generates 1000 samples from the test distribution and overlays the plot with pointwise interval estimates for the ordered quantiles from the test distribution.

Various option choices are available to effect different visualizations of the uncertainty surrounding the quantile quantile plot. These include overlaying independently generated exemplar test distribution sample quantile traces so as to assess the joint (as opposed to pointwise) distribution of quantiles.

See argument descriptions and examples for more details.

Usage

```
qqtest(
  data,
  dist = c("gaussian", "normal", "log-normal", "half-normal", "uniform", "exponential",
    "chi-squared", "kay", "student", "t"),
  df = 1,
  qfunction = NULL,
  rfunction = NULL,
  dataTest = NULL,
  p = NULL,
  a = NULL,
  np = NULL,
  matchMethod = c("hinges", "quartiles", "middlehalf", "bottomhalf", "tophalf"),
  xAxisAsProbs = FALSE,
  yAxisAsProbs = FALSE,
  xAxisProbs = c(0.05, 0.25, 0.5, 0.75, 0.95),
```

```
  yAxisProbs = c(0.05, 0.25, 0.5, 0.75, 0.95),
  nreps = 1000,
  centralPercents = c(0.9, 0.95, 0.99),
  envelope = TRUE,
  drawPercentiles = FALSE,
  drawQuartiles = FALSE,
  legend = NULL,
  legend.xy = "topleft",
  legend.cex = 0.8,
  nexemplars = 0,
  typex = NULL,
  plainTrails = FALSE,
  colTrails = NULL,
  alphaTrails = 0.25,
  lwdTrails = 1,
  lineup = FALSE,
  nsuspects = 20,
  col = NULL,
  h = 260,
  c = 90,
  l = 60,
  alpha = 1,
  cex = NULL,
  pch = 19,
  type = NULL,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  xlim = NULL,
  ylim = NULL,
  axes = NULL,
  bty = "o",
  ...
)
```

Arguments

data	A univariate dataset to be tested. If data has more than one column, the first is used.
dist	The name of the distribution against which the comparison is made, the test distribution for a few built-in distributions. One of "gaussian" (or "normal"), "log normal","half normal", "uniform","exponential","student", "chi-squared", or "kay". Only the first three characters of any of these is needed to specify the dist. If dist is "student", "chi-squared", or "kay", then a value for the degrees of freedom argument (df below) is also required.
df	Degrees of freedom of dist to be used when dist is either "student" or "chi-squared".

qfunction — If non-NULL, this must be a function of a single argument (a proportion, p say) which will be used to calculate the quantiles for the test distribution. If non-NULL, the rfunction should also be non-NULL. The value of the dist argument will be ignored when this is the case.

rfunction — If non-NULL, this must be a function of a single argument (a count, n say) which will be used to randomly select a sample of size n from the test distribution. If non-NULL, the qfunction must also be non-NULL. If qfunction is non-NULL and rfunction is NULL, then qfunction will be applied to the output of a call to runif in place of the NULL rfunction (i.e. a probability integral transform is used to generate a random sample).

The value of the dist argument will be ignored whenever qfunction is a function.

dataTest — If non-NULL, this must be a second data set. The empirical distribution given by this data will be used as the test distribution against which the value of data will be tested. If non-NULL, the values of the arguments dist, qfunction, and rfunction will all be ignored in favour of using this empirical distribution as the test distribution.

p — If non-NULL, this must be a vector containing the probability points at which the quantiles are to be calculated. If the length of this vector is the same as that of the data, then p is taken to be the probabilities that correspond to the data; otherwise the data are taken to provide and empirical quantile function values of which are taken at p to produce the plot.

If p is NULL (the default), then p is determined by the function ppoints(n,a) where n is the number of data points in data and a is given by the argument a below.

a — This is the second parameter given to the ppoints call to determine p as necessary. If a is NULL, then default values are chosen for each specific distribution (e.g. 3/8 for gaussian, 0 for uniform, etc.) and as 2/5 otherwise, following the recommendations of C. Cunnane (1978). Users may supply their own value such as the original Hazen's 1/2 or Tukey's 1/3.

np — This is required if the vector p is provided. Because p could take any values, we need to know np the sample size that was used to construct the quantiles at the provided vlaues p. When p and np are provided all simulation is based on simulating values from the distribution of the order statistics p*np.

matchMethod — It is necessary to match locations and scales of the two distributions. The method used to do this matching is given by matchMethod which must be one of c("hinges","quartiles", "middlehalf", "bottomhalf", "tophalf"). A line is fitted to the data and its coefficients used to adjust the location and scale of the test quantiles to match the data.

If matchMethod = "hinges" (the default), then the line is fit to (x, y) pairs given by the three central values of fivenum() on each of the coordinates; if "quartiles" the three quartiles of each coordinate are paired and a line fit. The method "middlehalf", as the name suggests, fits a line to the middle half of the sorted data. In this way, it is very similar to the previous two methods. All three are fairly robust and use the central part of the data to estimate the location and scale based roughly on matching medians and inter-quartile spreads.

The remaining two methods, "bottomhalf" and "tophalf", use a line fitted to the bottom and top half of the sorted data. These are not generally recommended but might be used when the comparison dist is skewed. For example with a chi-squared distribution, one might use bottomhalf to match on data from the shorter left tail of the distribution. Similarly, "tophalf" might be preferred when dist is skewed in the opposite direction.

Finally, note that the user may supply their own matching method by providing a function having vector arguments x and y for the horizontal and vertical quantiles which returns a vector of coefficients (intercept, slope) for the line to be used to match the quantile location and scales.

xAxisAsProbs — If TRUE (the default is FALSE) the horizontal axis will be labelled as probabilities. These are the cumulative probabilities according to the test distribution. They are located at the corresponding quantile values. They are handy in comparing percentiles of the test and data distributions as well as giving some measure of the symmetry and tail weights of the test distribution by their location. If FALSE the axis is labelled according to the quantile values.

yAxisAsProbs — If TRUE (the default is FALSE) the vertical axis will be labelled as probabilities. These are the cumulative probabilities according to the empirical distribution of the data. They are located at the corresponding quantile values. They are handy in comparing percentiles of the test and data distributions as well as giving some measure of the symmetry and tail weights of the data distribution by their location. If FALSE the axis is labelled according to the quantile values.

xAxisProbs — A vector of probabilities to be used to label the x axis ticks when xAxisAsProbs is TRUE. Default is c(0.05, 0.25, 0.50, 0.75, 0.95). Ignored if xAxisAsProbs is FALSE.

yAxisProbs — A vector of probabilities to be used to label the y axis ticks when yAxisAsProbs is TRUE. Default is c(0.05, 0.25, 0.50, 0.75, 0.95). Ignored if yAxisAsProbs is FALSE.

nreps — The number of replicate samples to be taken from the test distribution to construct the pointwise intervals for each quantile. Default is 1000. From these samples, an empirical distribution is generated from the test distribution for the ordered quantiles corresponding to the values ofppoints(length(data)). These are used to construct central intervals of whatever proportions are given by centralPercents.

centralPercents — The vector of proportions determining the central intervals of the empirical distribution of each ordered quantile from the test distribution. Default is c(0.90, 0.95, 0.99) corresponding to central 90, 95, and 99% simulated pointwise confidence intervals for each quantile coming from the test distribution for a sample the same size as data. The quality of these interval locations typically increases with nreps and decreases with the probability used for each interval.

envelope — If TRUE (the default), a grey envelope is plotted showing the central intervals for each quantile as a shade of grey. The higher is the corresponding probability associated with the interval, the lighter is the shade. The outermost edges of the envelope are the range of the simulated data from the test distribution. The envelope thus provides a (pointwise) density estimate of the quantiles drawn from the test distribution for this sample size. If FALSE no envelope is drawn.

drawPercentiles
: If TRUE, a pair of curves is plotted to show each of the central intervals as a different line type. These are plotted over the envelope if envelope is TRUE. If FALSE (the default) no simulated percentile curves are drawn.

drawQuartiles
: If TRUE, a pair of curves is plotted to show the quartiles (central 50% region) of the ordered quantiles simulated from the test distribution. The median of these is also plotted as a solid line type. These are plotted over the envelope if envelope is TRUE. If FALSE (the default) none of these curves are drawn.

legend
: If TRUE (the default is NULL) with nreps>0 a legend for the appearance of the simulated ranges of the central intervals is added to the plot. If FALSE, no legend appears. If NULL, legend always appears except when lineup = TRUE.

legend.xy
: Either a string being one of c("bottomright", "bottom", "bottomleft", "left", "topleft","top", "topright", "right", "center") (default is "topleft") or a list with named components x and y to be interpreted exactly as in the function legend() from the graphics package.

legend.cex
: Character expansion size for legend (default 0.8).

nexemplars
: (default is 0) The number of replicate samples to be taken from the test distribution and plotted as a coloured trail on the qqplot. Each such trail is a sample of the same size as data but truly coming from the test distribution. Each trail gives some idea of what the shape of a qqplot would be for a sample of that size from the test distribution. Together, they give some sense of the variability in the plot's shape.

typex
: (default is "o", or match type if supplied) The plot type to be used in the plotting of the exemplars, legal values are the same as for the type argument of plot.

plainTrails
: If TRUE, then a single grey colour is used for all exemplar trails. If FALSE (the default), each exemplar trail is shown in a different colour.

colTrails
: Colours to be used for the trails (default will be multi-colours).

alphaTrails
: The alpha transparency to be used in plotting all exemplar trails. The default is 0.25. Because the trails will overplot, a smaller alphaTrails value is recommended as nexemplars increases.

lwdTrails
: The graphical line width (lwd) to be used in plotting all exemplar trails. The default is 1. Because the trails will overplot, combining a larger lwdTrails with envelope = FALSE, a lower alphaTrails value larger nexemplars can give a truer sense of the density of qqplot configurations than with envelope = TRUE.

lineup
: If TRUE (default is FALSE) the qqplot of data is randomly located in a grid of nsuspects plots. Identical arguments are given to construct all qqtest plots in the grid. Assuming the viewer has not seen the qqplot of this data before, a successful selection of the true data plot out of the grid of plots corresponds to evidence against the hypothesis that the data come from the test distribution. Significance level is 1/nsuspects. Similarly, if the viewer narrows the selection down to only k possible subjects AND the true location is among them, then the corresponding observed level of significance (p-value) would be k//nsuspects.

  Each plot is given a suspect number from 1 to nsuspects (left to right, top to bottom). The suspect number of the plot corresponding to the actual data is returned, slightly obfuscated to help keep the test honest.

`nsuspects` The total number of plots (default is 20) to be viewed in the lineup display when `lineup` is `lineup`.

`col` If non-`NULL`, `col` must be colour to be used for the points in the plot. If `NULL` (the default), an `hcl` colour will be used from the values of the arguments `h`, `c`, `l`, and `alpha`.

`h` The hue of the colour of the points. Specified as an angle in degrees from 0 to 360 around a colour wheel. E.g. 0 is red, 120 green, 240 blue, Default is 260 (a bluish).

`c` The chroma of the colour of the points. Takes values from 0 to an upper bound that is a function of hue, `h`, and luminance, `l`. Roughly, for fixed `h` and `l` the higher the value of `c` the greater the intensity of colour.

`l` The luminance of the colour of the points. Takes values from 0 to 100. For any given combination of hue, `h`, and chroma, `c`, only a subset of this range will be possible. Roughly, for fixed `h` and `c` the higher the value of `l` the lighter is the colour.

`alpha` The alpha transparency of the colour of the points. Takes values from 0 to 1. Values near 0 are more transparent, values near 1 (the default) are more opaque. Alpha values sum when points over plot, giving some indication of density.

`cex` The graphical parameter `cex` for the size of the points.

`pch` The graphical parameter `pch` for the point character to be used for the points. Default is 19, a filled circle.

`type` The graphical parameter `type` for the points of the qqplot. Default is "p".

`main` The graphical parameter `main` providing a title for the plot. If `NULL` (the default), the title will be "qqtest" when `lineup = FALSE` and "Suspect: " followed by the plot number when `lineup = TRUE`. An empty string will suppress the title.

`xlab` The graphical parameter `xlab` labelling the x axis of the plot. If `NULL` (the default), an `xlab` is created based on the information available from the other arguments to `qqtest` about the test distribution. An empty string will suppress the labelling.

`ylab` The graphical parameter `ylab` labelling the y axis of the plot. If `NULL` (the default), a `ylab` is created based on the information available from the other arguments to `qqtest`. An empty string will suppress the labelling.

`xlim` The graphical parameter `xlim` determining the display limits of the x axis.

`ylim` The graphical parameter `ylim` determining the display limits of the y axis.

`axes` The graphical parameter `axes` determining whether axes are to be displayed (default is `NULL` which the same as `TRUE` except when `lineup=TRUE`, then `axes` is `FALSE`).

`bty` The graphical parameter `bty` determining the type of box to be used to enclose the plot (default is `"o"`, set `bty="n"` for no box).

`...` Any further graphical parameters to be passed to the `plot` function.

Value

Displays the qqplot.

Invisibly returns a list with named components `x`, `y`, and `order` giving the horizontal and vertical locations of the points in sorted order, as well as the order vector from the input data. The result of `qqtest` must be assigned to get these. The values could then, for example, be used to identify or label points in the display.

When `lineup` is `TRUE`, it returns a string encoding the true location of the data as a calculation to be evaluated. This provides some simple obfuscation of the true location so that the visual assessment can be honest. The true location is revealed by calling `revealLocation()` on the returned value.

Source

"Self calibrating quantile-quantile plots", R. Wayne Oldford, The American Statistician, 70, (2016) https://doi.org/10.1080/00031305.2015.1090338

"Unbiased Plotting Positions – A Review", C. Cunnane, Journal of Hydrology, Vol. 37 (1978), pp. 205-222.

Examples

```
#
# default qqtest plot
qqtest(precip, main = "Precipitation (inches/year) in 70 US cities")
#
# qqtest to compare to qqnorm
op <- par(mfrow=c(1,2))
qqnorm(precip, main="qqnorm")
qqtest(precip, main="qqtest",
       xAxisAsProbs=FALSE, yAxisAsProbs=FALSE)
par(op)
#
#  Use lines instead of envelope
qqtest(precip, envelope=FALSE, drawPercentiles=TRUE,
       main = "Precipitation (inches/year) in 70 US cities")
#
#  Use quartiles instead of envelope
qqtest(precip, envelope=FALSE, drawQuartiles=TRUE,
       main = "Precipitation (inches/year) in 70 US cities")
#
#  Use coloured exemplars (qqplot of data simulated from the test distribution)
#  and suppress the envelope.  Where the envelope, percentiles, and quartiles are
#  simulated pointwise bands, exemplars give some sense of what the (joint) shape of the
#  quantile-quantile plot should look like (for data from the test distribution).
#  Each simulated sample is a different colour.
qqtest(precip, nexemplars=10, typex="o", envelope=FALSE, type="p",
       main = "Precipitation (inches/year) in 70 US cities")
#
#  Alternatively, the trail of each exemplar could be plain (the identical grey).
#  Making each trail wide and assigning it some transparency (alpha near 0)
#  allows the trails to give a sense of the density through the darkness of the grey.
#
```

```
qqtest(precip, nexemplars=20, envelope=FALSE,
       lwdTrails=3, plainTrails=TRUE, alphaTrail=0.4, typex="o", type="o",
       main = "Precipitation (inches/year) in 70 US cities")
#
#  Wide coloured exemplars with some transparency provide an indication of
#  density and allow some trails to be followed by colour.
#
qqtest(precip, nexemplars=20, envelope=FALSE,
       lwdTrails=3,  alphaTrail=0.4, typex="o", type="o", col="black",
       main = "Precipitation (inches/year) in 70 US cities")


#  Envelope and exemplars with coloured trails to be followed.
#
qqtest(precip, nexemplars=5,
       lwdTrails=2, alphaTrail=0.6, alpha=0.8,
       main = "Precipitation (inches/year) in 70 US cities")
#
#
#  gaussian - qqplot, but now showing in the line up
trueLoc <- qqtest(precip, lineup=TRUE, main="Suspect", legend=FALSE,
                  cex=0.75, col="grey20", ylab="", pch=21)
# the location of the real data in the line up can be found by evaluating
# the contents of the string
trueLoc
#
# Cut and paste the string contents into the R console, or simply
revealLocation(trueLoc)
#
#
# log-normal ... using the bacteria data from Whipple(1916)
data(bacteria, package="qqtest")
# Note that these are selected percentiles from a sample of 365 days in a year
with(bacteria,
    qqtest(count, dist = "log-normal", p=percentTime/100, np=365, type="o",
     yAxisAsProbs=FALSE, ylab="bacteria per cc",
           xAxisProbs = c(0.01, 0.50,0.75, 0.90, 0.95, 0.99, 0.995),
           xlab="Percentage of days in 1913",
           main = "Number of bacteria from the Delaware river in 1913")
    )
ptics <- c(0.01, 0.10, 0.25, 0.50, 0.75, 0.90, 0.99 )
axis(1,at=qnorm(ptics), labels=floor(ptics*100))
yvals <- c(100, 1000, 10000, 100000)
axis(2, at=log(yvals,10),
     labels=c("100", "1,000", "10,000", "100,000"))
#
# compare this to the log-scaled normal qqplot
#
#
with(bacteria,
    qqtest(log(count, 10), dist = "normal",
           p=percentTime/100, np=365,
     type="o", axes=FALSE,
```

```
           ylab="bacteria per cc",
           xlab="Proportion of days in 1913",
           main = "Number of bacteria from the Delaware river in 1913")
    )
#
#
# Half normal ... using the penicillin data from Daniel(1959)
data(penicillin)

qqtest(penicillin, dist = "half-normal")

# Or the same again but with significant contrast labelled


with (penicillin,
 {qqtest(value, yAxisProbs=c(0.1, 0.75, 0.90, 0.95),
         dist="half-normal",
   ylab="Sample cumulative probability",
         xlab="Half-normal cumulative probability")
  ppAdj <- (1+ppoints(31))/2  # to get half-normals from normal
  x <- qnorm(ppAdj)
  valOrder <- order(value)    # need data and rownames in increasing order
  y <- value[valOrder]
  tags <- rownames(penicillin)[valOrder]
  selPoints <- 28:31          # going to label only the largest effects
  xoffset <- c(0.01, 0.02, 0.03, 0.075)  # text function is a bit off
  text(x[selPoints]-xoffset, y[selPoints],
       tags[selPoints],
       pos=2, cex=0.75)
 }
)

# Alternatively, use the returned results for a lot less work
results <- qqtest(penicillin$value, yAxisProbs=c(0.1, 0.75, 0.90, 0.95),
                    dist="half-normal",
                    ylab="Sample cumulative probability",
                    xlab="Half-normal cumulative probability")
tags <- row.names(penicillin)[results$order]
selPoints <- 28:31          # going to label only the largest effects
xoffset <- c(0.01, 0.02, 0.03, 0.075)  # text function is a bit off
text(results$x[selPoints]-xoffset, results$y[selPoints],
     tags[selPoints],
     pos=2, cex=0.75)

# or the same points could have been identified interactively vusing
# identify(results$x, results$y, labels = row.names(penicillin)[results$order])
#
# K on 9 df  ... see help(dkay)
# Use data on primer paint thickness (standard deviations on n=10)
data(primer, package="qqtest")
with (primer,
      qqtest(s,  dist="kay", df=9,
          yAxisAsProbs=FALSE,
```

```
       ylab="Standard deviation of primer thickness (in mils)")
  )
#
# chi-squared on 3 df
# Use robust covariance matrix in calculation Mahalanobis distances
# for the classical Brownlee stackloss data.
data(stacklossDistances, package="qqtest")
with(stacklossDistances,
     qqtest(robust, dist="chi", df=3, ylab="Robust Mahalanobis distances"))
#
#
# user supplied qfunction and rfunction -- compare to beta distribution
qqtest(precip,
       qfunction=function(p){qbeta(p, 2, 2)},
       rfunction=function(n){rbeta(n, 2, 2)},
       main = "Precipitation (inches/year) in 70 US cities")
#
#
# user supplied qfunction only -- compare to beta distribution
qqtest(precip,
       qfunction=function(p){qbeta(p, 2, 2)},
       main = "Precipitation (inches/year) in 70 US cities")
#
# comparing data samples
#
# Does the sample of beaver2's temperatures look like they
# could have come from a distribution shaped like beaver1's?
#
  qqtest(beaver2[,"temp"],
         dataTest=beaver1[,"temp"],
         ylab="Beaver 2", xlab="Beaver 1",
         main="Beaver body temperatures")
```

`revealLocation` `revealLocation` *Revealing locations encoded as a string calculation.*

Description

Reveals the location by parsing and evaluating the string.

Usage

```
revealLocation(hiddenLocation)
```

Arguments

`hiddenLocation`	A character vector of calculation strings.

Value

Returns the value of the each calculation.

See Also

hideLocation

Examples

```
trueLoc <- hideLocation(3,100)
trueLoc
revealLocation(trueLoc)

n <- 200
trueLoc <- sample(1:n, 3)
trueLoc
ans <- hideLocation(trueLoc , n)
ans
revealLocation(ans)
```

---

rkay rkay *The K distribution - generating pseudo-random values*

Description

Random generation for the K distribution on df degrees of freedom having non-centrality parameter ncp.

A K distribution is the square root of a chi-square divided by its degrees of freedom. That is, if x is chi-squared on m degrees of freedom, then y = sqrt(x/m) is K on m degrees of freedom. Under standard normal theory, K is the distribution of the pivotal quantity s/sigma where s is the sample standard deviation and sigma is the standard deviation parameter of the normal density. K is the natural distribution for tests and confidence intervals about sigma. K densities are more nearly symmetric than are chi-squared and concentrate near 1. As the degrees of freedom increase, they become more symmetric, more concentrated, and more nearly normally distributed.

Usage

```
rkay(n, df, ncp = 0)
```

Arguments

n	Number of observations. If length(n) > 1, the length is taken to be the number required.
df	Degrees of freedom (non-negative, but can be non-integer).
ncp	Non-centrality parameter (non-negative).

Value

rkay returns pseudo-randomly generated values.

Invalid arguments will result in return value NaN, with a warning.

Note

Depends on call to analogous chi-squared functions. See rchisq for details on non-centrality parameter calculations.

Examples

```
x <- rkay(100, 20)
hist(x, main="100 observations from a K(20)")
# Certainly looks like it comes from a K on 20
qqtest(x, dist="kay",df=20)
# for this many degrees of freedom it looks
# a lot like a gaussian (normal) distribution
qqtest(x, dist="gau",df=1)
# But not like it came from a K on 1 degree of freedom
qqtest(x, dist="kay",df=1)
#
# See the vignette for more on the "K-distribution"
#
```

---

sittingHeights *Sitting height in inches of female adults (aged 23-50).*

Description

From measurements made by Francis Galton at London's International Health Exhibition in 1884, published in 1885.

Usage

```
sittingHeights
```

Format

A data frame with 9 rows and 8 variates, the first 5 of which are as recorded by Galton:

lowerBound Sitting height greater than or equal to this lower bound in inches.

upperBound Sitting height strictly less than this upper bound in inches.

nCases Number of cases observed with sitting height between the two bounds.

nCasesCumulative Number of cases observed with sitting height up to but not including the upper bound.

percentCumulative Cumulative number of cases expressed as a percent.

binCentre Average of the lower and upper bounds.

nCasesCentred Number of cases assigned to the binCentre; half of cases observed with sitting height between the two bounds is assigned to be below the binCentre, half above (avoids producing 100 percent for last entry).

proportionCumulativeAdjusted Cumulative proportions using nCasesCentred.

Details

`with(sittingHeights, plot(percentCumulative,upperBound, type="o", lwd=2, xlim=c(0,100), ylim=c(20,40),xlab="Percentage of women", ylab="Sitting heights in inches"))` will effect Galton's Ogive.

`with(sittingHeights, qqtest(binCentre, dist="normal", p = proportionCumulativeAdjusted, np=775, main="Sitting heights of women in inches"))` will effect a normal qqplot for this data.

Source

"The Application of a Graphic Method to Fallible Measures", Francis Galton, (1885), Journal of the Statistical Society of London, Jubilee Volume (June 22-24, 1885), pp. 262-265.

`stacklossDistances` *Mahalanobis squared distances of Brownlee's stack loss plant operation data based only on the explanatory variates (air flow, water temperature, and acid concentration).*

Description

Mahalanobis distances were calculated using the mahalanobis R function. Under standard normal theory, these are approximately Chi-squared on 3 degrees of freedom.

Usage

```
stacklossDistances
```

Format

A data frame with 21 rows and 2 variates:

ordinary Mahalanobis squared distances from the arithemetic mean using the eliptical contours of the sample covariance matrix.

robust As with distances, these are Mahalobis squared distances butnow based on robust measures of location and covariance matrix (as determined from the default covRob of the robust package).

Details

`with(stacklossDistances, qqtest(robust,dist="chi", df=3))` will show "outliers".

`with(stacklossDistances, qqtest(ordinary,dist="chi", df=3))` will show "inliers".

Source

"Statistical Theory and Methodology in Science and Engineering", K.A. Brownlee, (1960, 2nd ed. 1965), Wiley, New York pp. 491-500.

`WachusettReservoir` *Storage, in millions of gallons daily per square mile of net land area, at the Wachusett Reservoir in Massacusetts - storage computed for each of several rates of draft (draft being a determined maintainable flow in 1,000s of gallons per square mile daily).*

Description

First extensive published use of normal qqplots. Hazen uses a=1/2 to make the p values for the plots. Hazen doesn'tplot zeros but has them contribute to the sample size. The context of use is in a study of the relation between the water storage provided in a reservoir on any stream and the quantity of water that can be continuously supplied by it. To quote the paper: ... treat all the remaining variations on the basis of probabilities, using all data from a number of streams; and to study them in comparison with the normal law of error."

Usage

```
WachusettReservoir
```

Format

A data frame with 15 rows and 6 variates:

draft100 Computed storage, in millions of gallons per square mile of land area, given a draft of 100,000 gallons per square mile daily.

draft200 Computed storage, in millions of gallons per square mile of land area, given a draft of 200,000 gallons per square mile daily.

draft400 Computed storage, in millions of gallons per square mile of land area, given a draft of 400,000 gallons per square mile daily.

draft600 Computed storage, in millions of gallons per square mile of land area, given a draft of 600,000 gallons per square mile daily.

draft800 Computed storage, in millions of gallons per square mile of land area, given a draft of 800,000 gallons per square mile daily.

draft1000 Computed storage, in millions of gallons per square mile of land area, given a draft of 1,000,000 gallons per square mile daily.

Details

`qqtest(WachusettReservoir$draft800,dist="uniform", a=1/2,type="o")` will effect Hazen's original plot for a draft of 800,000 gallons per square mile daily.

`qqtest(WachusettReservoir$draft800,dist="normal", a=1/2, type="o")` will effect Hazen's normal qq plot for a draft of 800,000 gallons per square mile daily.

Source

"Storage to be provided in impounding reservoirs for municipal water supply (with discussion)", Allen Hazen, Transactions of the American Society of Civil Engineers, Vol. 77, (1914), pp. 1539-1669.

Index