Package ‘qqplotr’

April 23, 2021

Type Package
Version 0.0.5
Title Quantile-Quantile Plot Extensions for 'ggplot2'
Description Extensions of 'ggplot2' Q-Q plot functionalities.
URL https://github.com/aloy/qqplotr

BugReports https://github.com/aloy/qqplotr/issues
License GPL-3 | file LICENSE
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1

Collate 'data.R' 'geom_qq_band.R' 'qqplotr.R' 'runShinyExample.R'
            'stat_pp_band.R' 'stat_pp_line.R' 'stat_pp_point.R'
            'stat_qq_line.R' 'stat_qq_band.R' 'stat_qq_point.R'

VignetteBuilder knitr

Depends R (>= 3.1), ggplot2 (>= 2.2)
Imports dplyr, robustbase, MASS
Suggests shiny, devtools, lattice, shinyBS, knitr, rmarkdown

NeedsCompilation no

Author Alexandre Almeida [aut],
      Adam Loy [aut, cre],
      Heike Hofmann [aut]

Maintainer Adam Loy <loyad01@gmail.com>

Repository CRAN

Date/Publication 2021-04-23 14:30:02 UTC
**Description**

Draws quantile-quantile confidence bands, with an additional detrend option.

**Usage**

```r
geom_qq_band(
  mapping = NULL,
  data = NULL,
  stat = "qq_band",
  position = "identity",
  na.rm = TRUE,
  show.legend = NA,
  inherit.aes = TRUE,
  distribution = "norm",
  dparams = list(),
  detrend = FALSE,
  identity = FALSE,
  qtype = 7,
  qprobs = c(0.25, 0.75),
  bandType = "pointwise",
  B = 1000,
  conf = 0.95,
  mu = NULL,
  sigma = NULL,
  ...
)
```

```r
stat_qq_band(
  mapping = NULL,
  data = NULL,
```
geom_qq_band

geom = "qq_band",
position = "identity",
na.rm = TRUE,
show.legend = NA,
inherit.aes = TRUE,
distribution = "norm",
dparams = list(),
detrend = FALSE,
identity = FALSE,
qtype = 7,
qprobs = c(0.25, 0.75),
bandType = "pointwise",
B = 1000,
conf = 0.95,
mu = NULL,
sigma = NULL,
...
)

Arguments

mapping Set of aesthetic mappings created by aes() or aes(.). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.

data The data to be displayed in this layer. There are three options:
If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot().
A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created.
A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x,10)).

stat statistic to use to calculate confidence bands. Should be ‘qq_band’.

position Position adjustment, either as a string, or the result of a call to a position adjustment function.

na.rm If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.

show.legend logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.

inherit.aes If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn’t inherit behaviour from the default plot specification, e.g. borders().

distribution Character. Theoretical probability distribution function to use. Do not provide the full distribution function name (e.g., "dnorm"). Instead, just provide its
shortened name (e.g., "norm"). If you wish to provide a custom distribution, you may do so by first creating the density, quantile, and random functions following the standard nomenclature from the stats package (i.e., for "custom", create the dcustom, pcustom, qcustom, and rcustom functions).

dparams | List of additional parameters passed on to the previously chosen distribution function. If an empty list is provided (default) then the distributional parameters are estimated via MLE. MLE for custom distributions is currently not supported, so you must provide the appropriate dparams in that case.

detrend | Logical. Should the plot objects be detrended? If TRUE, the objects will be detrended according to the reference Q-Q line. This procedure was described by Thode (2002), and may help reducing visual bias caused by the orthogonal distances from Q-Q points to the reference line.

identity | Logical. Should an identity line be used as the reference line used to construct the confidence bands? If TRUE, the identity line is used. If FALSE (default), the commonly-used Q-Q line that intercepts two data quantiles specified in qprobs is used. Please notice that the chosen reference line will also be used for the detrending procedure, if detrend = TRUE.

qtype | Integer between 1 and 9. Type of the quantile algorithm to be used by the quantile function to construct the Q-Q line.

qprobs | Numeric vector of length two. Represents the quantiles used by the quantile function to construct the Q-Q line.

bandType | Character. Either "pointwise", "boot", "ks" or "ts". "pointwise" constructs pointwise confidence bands based on Normal confidence intervals. "boot" creates pointwise confidence bands based on a parametric bootstrap; parameters are estimated with MLEs. "ks" constructs simultaneous confidence bands based on the Kolmogorov-Smirnov test. Finally, "ts" constructs tail-sensitive confidence bands, as described by Aldor-Noiman et al. (2013) (also, see 'Note' for limitations).

B | Integer. If bandType = "boot", then B is the number of bootstrap replicates. If bandType = "ts", then B is the number of simulated samples.

conf | Numerical. Confidence level of the bands.

mu | Numerical. Only used if bandType = "ts". Center distributional parameter used to construct the simulated tail-sensitive confidence bands. If either mu or sigma are NULL, then those parameters are estimated using \( \hat{Q}_n \) and \( s_{\hat{Q}_n} \), respectively.

sigma | Numerical. Only used if bandType = "ts". Scale distributional parameter used to construct the simulated tail-sensitive confidence bands. If either mu or sigma are NULL, then those parameters are estimated using robust estimates from the stats package.

... | Other arguments passed on to layer(). These are often aesthetics, used to set an aesthetic to a fixed value, like colour = "red" or size = 3. They may also be parameters to the paired geom/stat.

geom | The geometric object to use display the data
Note

• Tail-sensitive confidence bands are only implemented for Normal Q-Q plots. As a future update, we intend to generalize to other distributions.

• Bootstrap bands are constructed based on a MLE parametric bootstrap. Hence, it is not possible to construct such bands if the sample and theoretical distributions present mismatching supports.

References


Examples

# generate random Normal data
set.seed(0)
smp <- data.frame(norm = rnorm(100))

# Normal Q-Q plot of Normal data
gg <- ggplot(data = smp, mapping = aes(sample = norm)) +
  stat_qq_band() +
  stat_qq_line() +
  stat_qq_point()
gg + labs(x = "Theoretical Quantiles", y = "Sample Quantiles")

# Exponential Q-Q plot of mean ozone levels (airquality dataset)
di <- "exp"
dp <- list(rate = 1)
gg <- ggplot(data = airquality, mapping = aes(sample = Ozone)) +
  stat_qq_band(distribution = di, dparams = dp) +
  stat_qq_line(distribution = di, dparams = dp) +
  stat_qq_point(distribution = di, dparams = dp) +
  labs(x = "Theoretical Quantiles", y = "Sample Quantiles")
gg

# Detrended Exponential Q-Q plot of mean ozone levels
di <- "exp"
dp <- list(rate = 1)
de <- TRUE
gg <- ggplot(data = airquality, mapping = aes(sample = Ozone)) +
  stat_qq_band(distribution = di, detrend = de) +
  stat_qq_line(distribution = di, detrend = de) +
  stat_qq_point(distribution = di, detrend = de) +
  labs(x = "Theoretical Quantiles", y = "Sample Quantiles")
gg

# Normal Q-Q plot of Normal data with bootstrap confidence bands
bt <- "boot"
gg <- ggplot(data = smp, mapping = aes(sample = norm)) +
  stat_qq_band(bandType = bt) +
```r
stat_qq_line() +
stat_qq_point() +
labs(x = "Theoretical Quantiles", y = "Sample Quantiles")

# Normal Q-Q plot of Normal data with tail-sensitive confidence bands
bt <- "ts"
gg <- ggplot(data = smp, mapping = aes(sample = norm)) +
      stat_qq_band(bandType = bt) +
      stat_qq_line() +
      stat_qq_point() +
      labs(x = "Theoretical Quantiles", y = "Sample Quantiles")
gg
```

---

**iowa**

*2012 BRFSS sample for the state of Iowa*

---

**Description**

2012 BRFSS sample for the state of Iowa

**Usage**

```r
data(iowa)
```

**Format**

A data frame with 7166 observations on 3 variables:

- **SEX** Gender
- **WTKG3** Weight in kg
- **HTIN4** Height in inch

**Source**

https://www.cdc.gov/brfss/annual_data/annual_2012.html
Description
Men’s Olympic Long Jump Qualifiers 2012

Usage
data(longjump)

Format
A data frame with 42 observations on the following 4 variables:
- **rank**  Athlete’s rank at the qualifying event
- **name**  Athlete’s name
- **country**  Athlete’s country of origin
- **distance**  Result in meters

Source
https://www.olympic.org/london-2012/athletics/long-jump-men

Description
This package extends some ggplot2 functionalities by permitting the drawing of both quantile-quantile (Q-Q) and probability-probability (P-P) points, lines, and confidence bands. The functions of this package also allow the detrend adjustment, proposed by Thode (2002), which helps reduce visual bias when assessing those plots.

Details
The functions of this package, presented as ggplot2 Stats, are divided into two groups: Q-Q and P-P related.

Each of the groups is composed of three Stats: **point**, **line**, and **band**. Those Stats, while independent, complement each other when plotted together.
stat_pp_band  Probability-probability confidence bands

Description

Draws probability-probability confidence bands.

Usage

stat_pp_band(
  mapping = NULL,
  data = NULL,
  geom = "ribbon",
  position = "identity",
  na.rm = TRUE,
  show.legend = NA,
  inherit.aes = TRUE,
  distribution = "norm",
  dparams = list(),
  bandType = "boot",
  B = 1000,
  conf = 0.95,
  detrend = FALSE,
  ...
)

Arguments

- **mapping** Set of aesthetic mappings created by `aes()` or `aes()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.

- **data** The data to be displayed in this layer. There are three options:
  - If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`.
  - A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created.
  - A function will be called with a single argument, the plot data. The return value must be a `data.frame`, and will be used as the layer data. A function can be created from a formula (e.g. `~ head(.x,10)`).

- **geom** The geometric object to use display the data

- **position** Position adjustment, either as a string, or the result of a call to a position adjustment function.

- **na.rm** If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed.
stat_pp_band

show.legend logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.

inherit.aes If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn’t inherit behaviour from the default plot specification, e.g. borders().

distribution Character. Theoretical probability distribution function to use. Do not provide the full distribution function name (e.g., "dnorm"). Instead, just provide its shortened name (e.g., "norm"). If you wish to provide a custom distribution, you may do so by first creating the density, quantile, and random functions following the standard nomenclature from the stats package (i.e., for "custom", create the dcust, pcust, qcust, and rcust functions).

dparams List of additional parameters passed on to the previously chosen distribution function. If an empty list is provided (default) then the distributional parameters are estimated via MLE. MLE for custom distributions is currently not supported, so you must provide the appropriate dparams in that case.

bandType Character. Only "boot" is available for now. "boot" creates pointwise confidence bands based on a bootstrap.

B Integer. If bandType = "boot", then B is the number of bootstrap replicates.

conf Numerical. Confidence level of the bands.

detrend Logical. Should the plot objects be detrended? If TRUE, the objects will be detrended according to the default identity P-P line. This procedure was described by Thode (2002), and may help reducing visual bias caused by the orthogonal distances from P-P points to the reference line.

... Other arguments passed on to layer(). These are often aesthetics, used to set an aesthetic to a fixed value, like colour = "red" or size = 3. They may also be parameters to the paired geom/stat.

Examples

# generate random Normal data
set.seed(0)
smp <- data.frame(norm = rnorm(100), exp = rexp(100))

# Normal P-P plot of Normal data
gg <- ggplot(data = smp, mapping = aes(sample = norm)) +
  stat_pp_band() +
  stat_pp_line() +
  stat_pp_point() +
  labs(x = "Probability Points", y = "Cumulative Probability")
gg

# Shifted Normal P-P plot of Normal data
dp <- list(mean = 1.5)
gg <- ggplot(data = smp, mapping = aes(sample = norm)) +
  stat_pp_band(dparams = dp) +
  stat_pp_line() +
  stat_pp_point(dparams = dp) +
stat_pp_line

Description

Draws a probability-probability line.

Usage

stat_pp_line(
  mapping = NULL,
  data = NULL,
  geom = "path",
  position = "identity",
  na.rm = TRUE,
  show.legend = NA,
  inherit.aes = TRUE,
  ab = c(0, 1),
  detrend = FALSE,
  ...
)

Arguments

mapping      Set of aesthetic mappings created by \texttt{aes()} or \texttt{aes()}. If specified and \texttt{inherit.aes} = \texttt{TRUE} (the default), it is combined with the default mapping at the top level of the plot. You must supply \texttt{mapping} if there is no plot mapping.
The data to be displayed in this layer. There are three options:

If NULL, the default, the data is inherited from the plot data as specified in the call to `ggplot()`.

A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created.

A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. `~ head(.x,10)`).

The geometric object to use display the data

Position adjustment, either as a string, or the result of a call to a position adjustment function.

If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.

logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.

If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn’t inherit behaviour from the default plot specification, e.g. `borders()`.

Numeric vector of length two. The intercept (a) and slope (b) of the P-P line. Defaults to the identity line (a = 0, b = 1).

Logical. Should the plot objects be detrended? If TRUE, the objects will be detrended according to the default identity P-P line. This procedure was described by Thode (2002), and may help reducing visual bias caused by the orthogonal distances from P-P points to the reference line.

Other arguments passed on to `layer()`. These are often aesthetics, used to set an aesthetic to a fixed value, like colour = "red" or size = 3. They may also be parameters to the paired geom/stat.

Examples

```r
# generate random Normal data
data_norm <- data.frame(norm = rnorm(100))

# Normal P-P plot of Normal data
gg_norm <- ggplot(data = data_norm, mapping = aes(sample = norm)) +
  stat_pp_line() +
  stat_pp_point() +
  labs(x = "Probability Points", y = "Cumulative Probability")
gg_norm

# Shifted Normal P-P plot of Normal data
data_shifted_norm <- data.frame(norm = rnorm(100) + 1.5)
dp_shifted_norm <- list(mean = 1.5)
gg_shifted_norm <- ggplot(data = data_shifted_norm, mapping = aes(sample = norm)) +
  stat_pp_line() +
```
stat_pp_point(dparams = dp) +
labs(x = "Probability Points", y = "Cumulative Probability")

# Normal P-P plot of mean ozone levels (airquality dataset)
dp <- list(mean = 38, sd = 27)
gg <- ggplot(data = airquality, mapping = aes(sample = Ozone)) +
  stat_pp_line() +
  stat_pp_point(dparams = dp) +
  labs(x = "Probability Points", y = "Cumulative Probability")

stat_pp_point

**stat_pp_point**

**Probability-probability points**

**Description**

Draws probability-probability points.

**Usage**

```r
stat_pp_point(
  mapping = NULL,
  data = NULL,
  geom = "point",
  position = "identity",
  na.rm = TRUE,
  show.legend = NA,
  inherit.aes = TRUE,
  distribution = "norm",
  dparams = list(),
  detrend = FALSE,
  ...
)
```

**Arguments**

- `mapping`: Set of aesthetic mappings created by `aes()` or `aes()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
- `data`: The data to be displayed in this layer. There are three options:
  - If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`.`data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created.
A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g., `~ head(.x,10)`).

**geom**

The geometric object to use display the data

**position**

Position adjustment, either as a string, or the result of a call to a position adjustment function.

**na.rm**

If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed.

**show.legend**

Logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display.

**inherit.aes**

If `FALSE`, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn’t inherit behaviour from the default plot specification, e.g., `borders()`.

**distribution**

Character. Theoretical probability distribution function to use. Do not provide the full distribution function name (e.g., "dnorm"). Instead, just provide its shortened name (e.g., "norm"). If you wish to provide a custom distribution, you may do so by first creating the density, quantile, and random functions following the standard nomenclature from the stats package (i.e., for "custom", create the dcustom, pcustom, qcustom, and rcustom functions).

**dparams**

List of additional parameters passed on to the previously chosen distribution function. If an empty list is provided (default) then the distributional parameters are estimated via MLE. MLE for custom distributions is currently not supported, so you must provide the appropriate dparams in that case.

**detrend**

Logical. Should the plot objects be detrended? If `TRUE`, the objects will be detrended according to the default identity P-P line. This procedure was described by Thode (2002), and may help reducing visual bias caused by the orthogonal distances from P-P points to the reference line.

... Other arguments passed on to `layer()`. These are often aesthetics, used to set an aesthetic to a fixed value, like `colour = "red"` or `size = 3`. They may also be parameters to the paired geom/stat.

**References**


**Examples**

```r
# generate random Normal data
set.seed(0)
smp <- data.frame(norm = rnorm(100))

# Normal P-P plot of Normal data
gg <- ggplot(data = smp, mapping = aes(sample = norm)) + 
    stat_pp_point() + 
    labs(x = "Probability Points", y = "Cumulative Probability")
gg
```
# Shifted Normal P-P plot of Normal data
dp <- list(mean = 1.5)
gg <- ggplot(data = smp, mapping = aes(sample = norm)) +
  stat_pp_point(dparams = dp) +
  labs(x = "Probability Points", y = "Cumulative Probability")
gg

# Normal P-P plot of mean ozone levels (airquality dataset)
dp <- list(mean = 38, sd = 27)
gg <- ggplot(data = airquality, mapping = aes(sample = Ozone)) +
  stat_pp_point(dparams = dp) +
  labs(x = "Probability Points", y = "Cumulative Probability")
gg

stat_qq_line

Quantile-quantile lines

Description

Draws a quantile-quantile line, with an additional detrend option.

Usage

stat_qq_line(
  mapping = NULL,
  data = NULL,
  geom = "path",
  position = "identity",
  na.rm = TRUE,
  show.legend = NA,
  inherit.aes = TRUE,
  distribution = "norm",
  dparams = list(),
  detrend = FALSE,
  identity = FALSE,
  qtype = 7,
  qprobs = c(0.25, 0.75),
  ...
)

Arguments

mapping Set of aesthetic mappings created by aes() or aes(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
stat_qq_line

**data**
The data to be displayed in this layer. There are three options:
If NULL, the default, the data is inherited from the plot data as specified in the call to `ggplot()`.
A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created.
A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. `~ head(.x, 10)`).

**geom**
The geometric object to use display the data

**position**
Position adjustment, either as a string, or the result of a call to a position adjustment function.

**na.rm**
If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.

**show.legend**
Logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.

**inherit.aes**
If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn’t inherit behaviour from the default plot specification, e.g. `borders()`.

**distribution**
Character. Theoretical probability distribution function to use. Do not provide the full distribution function name (e.g., "dnorm"). Instead, just provide its shortened name (e.g., "norm"). If you wish to provide a custom distribution, you may do so by first creating the density, quantile, and random functions following the standard nomenclature from the stats package (i.e., for "custom", create the `dcustom`, `pcustom`, `qcustom`, and `rcustom` functions).

**dparams**
List of additional parameters passed on to the previously chosen distribution function. If an empty list is provided (default) then the distributional parameters are estimated via MLE. MLE for custom distributions is currently not supported, so you must provide the appropriate `dparams` in that case.

**detrend**
Logical. Should the plot objects be detrended? If TRUE, the objects will be detrended according to the reference Q-Q line. This procedure was described by Thode (2002), and may help reducing visual bias caused by the orthogonal distances from Q-Q points to the reference line.

**identity**
Logical. Should an identity line be used as the reference line? If TRUE, the identity line is used. If FALSE (default), the commonly-used Q-Q line that intercepts two data quantiles specified in `qprobs` is used. Please notice that the chosen reference line will also be used for the detrending procedure, if `detrend = TRUE`.

**qtype**
Integer between 1 and 9. Only used if `detrend = TRUE` and `identity = FALSE`. Type of the quantile algorithm to be used by the `quantile` function to construct the Q-Q line.

**qprobs**
Numeric vector of length two. Only used if `detrend = TRUE` and `identity = FALSE`. Represents the quantiles used by the `quantile` function to construct the Q-Q line.
... Other arguments passed on to `layer()`. These are often aesthetics, used to set an aesthetic to a fixed value, like `colour = "red"` or `size = 3`. They may also be parameters to the paired geom/stat.

References


Examples

```r
# generate random Normal data
set.seed(0)
smp <- data.frame(norm = rnorm(100))

# Normal Q-Q plot of Normal data
gg <- ggplot(data = smp, mapping = aes(sample = norm)) +
    stat_qq_line() +
    stat_qq_point() +
    labs(x = "Theoretical Quantiles", y = "Sample Quantiles")
gg

# Exponential Q-Q plot of mean ozone levels (airquality dataset)
di <- "exp"
dp <- list(rate = 1)
gg <- ggplot(data = airquality, mapping = aes(sample = Ozone)) +
    stat_qq_line(distribution = di, dparams = dp) +
    stat_qq_point(distribution = di, dparams = dp) +
    labs(x = "Theoretical Quantiles", y = "Sample Quantiles")
gg

# Detrended Exponential Q-Q plot of mean ozone levels
di <- "exp"
dp <- list(rate = 1)
de <- TRUE
gg <- ggplot(data = airquality, mapping = aes(sample = Ozone)) +
    stat_qq_line(distribution = di, detrend = de) +
    stat_qq_point(distribution = di, detrend = de) +
    labs(x = "Theoretical Quantiles", y = "Sample Quantiles")
gg
```

---

<table>
<thead>
<tr>
<th>stat_qq_point</th>
<th>Quantile-quantile points</th>
</tr>
</thead>
</table>

**Description**

Draws quantile-quantile points, with an additional detrend option.
stat_qq_point

Usage

stat_qq_point(
  mapping = NULL,
  data = NULL,
  geom = "point",
  position = "identity",
  na.rm = TRUE,
  show.legend = NA,
  inherit.aes = TRUE,
  distribution = "norm",
  dparams = list(),
  detrend = FALSE,
  identity = FALSE,
  qtype = 7,
  qprobs = c(0.25, 0.75),
  ...
)

Arguments

mapping Set of aesthetic mappings created by aes() or aes_.() If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.

data The data to be displayed in this layer. There are three options:
If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot().
A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created.
A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x,10)).

geom The geometric object to use display the data

position Position adjustment, either as a string, or the result of a call to a position adjust-
ment function.

na.rm If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.

show.legend logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.

inherit.aes If FALSE, overrides the default aesthetics, rather than combining with them.
This is most useful for helper functions that define both data and aesthetics and shouldn’t inherit behaviour from the default plot specification, e.g. borders().

distribution Character. Theoretical probability distribution function to use. Do not provide the full distribution function name (e.g., "dnorm"). Instead, just provide its shortened name (e.g., "norm"). If you wish to provide a custom distribution, you
may do so by first creating the density, quantile, and random functions following
the standard nomenclature from the stats package (i.e., for "custom", create
the dcustom, pcustom, qcustom, and rcustom functions).

- **dparams** List of additional parameters passed on to the previously chosen distribution
  function. If an empty list is provided (default) then the distributional parameters
  are estimated via MLE. MLE for custom distributions is currently not supported,
  so you must provide the appropriate dparams in that case.

- **detrend** Logical. Should the plot objects be detrended? If TRUE, the objects will be
detrended according to the reference Q-Q line. This procedure was described
by Thode (2002), and may help reducing visual bias caused by the orthogonal
distances from Q-Q points to the reference line.

- **identity** Logical. Only used if detrend = TRUE. Should an identity line be used as the
  reference line for the plot detrending? If TRUE, the points will be detrended
  according to the reference identity line. If FALSE (default), the commonly-used
  Q-Q line that intercepts two data quantiles specified in aprobs is used.

- **qtype** Integer between 1 and 9. Only used if detrend = TRUE and identity = FALSE.
  Type of the quantile algorithm to be used by the quantile function to construct
  the Q-Q line.

- **qprobs** Numeric vector of length two. Only used if detrend = TRUE and identity = FALSE.
  Represents the quantiles used by the quantile function to construct the
  Q-Q line.

- **...** Other arguments passed on to layer(). These are often aesthetics, used to set
  an aesthetic to a fixed value, like colour = "red" or size = 3. They may also
  be parameters to the paired geom/stat.

References


Examples

```r
# generate random Normal data
set.seed(0)
smp <- data.frame(norm = rnorm(100))

# Normal Q-Q plot of simulated Normal data
gg <- ggplot(data = smp, mapping = aes(sample = norm)) +
    stat_qq_point() +
    labs(x = "Theoretical Quantiles", y = "Sample Quantiles")
gg

# Exponential Q-Q plot of mean ozone levels (airquality dataset)
di <- "exp"
dp <- list(rate = 1)
gg <- ggplot(data = airquality, mapping = aes(sample = Ozone)) +
    stat_qq_point(distribution = di, dparams = dp) +
    labs(x = "Theoretical Quantiles", y = "Sample Quantiles")
gg
```
Index

* datasets
  iowa, 6
  longjump, 7
aes(), 3, 8, 10, 12, 14, 17
aes_(), 3, 8, 10, 12, 14, 17
borders(), 3, 9, 11, 13, 15, 17
fortify(), 3, 8, 11, 12, 15, 17
geom.qq_band, 2
ggplot(), 3, 8, 11, 12, 15, 17
iowa, 6
layer(), 4, 9, 11, 13, 16, 18
longjump, 7
Qn, 4
qqplotr, 7
quantile, 4, 15, 18
s_Qn, 4
stat.pp_band, 8
stat.pp_line, 10
stat.pp_point, 12
stat.qq_band (geom.qq_band), 2
stat.qq_line, 14
stat.qq_point, 16