Package ‘qqboxplot’

March 25, 2022

Title Implementation of the Q-Q Boxplot

Version 0.2.0

Description A system to implement the Q-Q boxplot. It is implemented as an extension to 'ggplot2'. The Q-Q boxplot is an amalgam of the boxplot and the Q-Q plot and allows the user to rapidly examine summary statistics and tail behavior for multiple distributions in the same pane. As an extension of the 'ggplot2' implementation of the boxplot, possible modifications to the boxplot extend to the Q-Q boxplot.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Imports ggplot2, grid

Depends R (>= 3.3)

Suggests knitr, rmarkdown, dplyr, gridExtra, testthat (>= 3.0.0), vdiffr (>= 0.3.3), scales

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Jordan Rodu [aut, cre]

Maintainer Jordan Rodu <jordan.rodu@gmail.com>

Repository CRAN

Date/Publication 2022-03-25 08:00:05 UTC

R topics documented:

`comparison_dataset` *Simulated normal dataset with mean=5 and variance=1*

Description

A dataset that contains simulated data to reproduce a figure in our manuscript

Usage

```
comparison_dataset
```

Format

A vector

Source

simulations

`expression_data` *Log expression data for select genes*

Description

A dataset that contains log expression data for randomly selected genes for two patients, one with autism and one control.

Usage

```
expression_data
```

Format

A data frame with 1200 rows and 3 variables:

gene gene identifier (not meaningful)

specimen autism or control

log_count the logged gene expression count ...

Source

https://www.ebi.ac.uk/gxa/experiments/E-GEOD-30573/Results

`geom_qqboxplot` *A modification of the boxplot with information about the tails*

Description

A modification of the boxplot with information about the tails

Usage

```
geom_qqboxplot(
  mapping = NULL,
  data = NULL,
  stat = "qqboxplot",
  position = "dodge2",
  ...,
  outlier.colour = NULL,
  outlier.color = NULL,
  outlier.fill = NULL,
  outlier.shape = 19,
  outlier.size = 1.5,
  outlier.stroke = 0.5,
  outlier.alpha = NULL,
  notch = FALSE,
  notchwidth = 0.5,
  varwidth = FALSE,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

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` Use to override the default connection between `geom_boxplot()` and `stat_boxplot()`.

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

`...` 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.

`outlier.colour` Default aesthetics for outliers. Set to `NULL` to inherit from the aesthetics used for the box.

In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.

Sometimes it can be useful to hide the outliers, for example when overlaying the raw data points on top of the boxplot. Hiding the outliers can be achieved by setting `outlier.shape = NA`. Importantly, this does not remove the outliers, it only hides them, so the range calculated for the y-axis will be the same with outliers shown and outliers hidden.

`outlier.color` Default aesthetics for outliers. Set to `NULL` to inherit from the aesthetics used for the box.

In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.

Sometimes it can be useful to hide the outliers, for example when overlaying the raw data points on top of the boxplot. Hiding the outliers can be achieved by setting `outlier.shape = NA`. Importantly, this does not remove the outliers, it only hides them, so the range calculated for the y-axis will be the same with outliers shown and outliers hidden.

`outlier.fill` Default aesthetics for outliers. Set to `NULL` to inherit from the aesthetics used for the box.

In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.

Sometimes it can be useful to hide the outliers, for example when overlaying the raw data points on top of the boxplot. Hiding the outliers can be achieved by setting `outlier.shape = NA`. Importantly, this does not remove the outliers, it only hides them, so the range calculated for the y-axis will be the same with outliers shown and outliers hidden.

`outlier.shape` Default aesthetics for outliers. Set to `NULL` to inherit from the aesthetics used for the box.

In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.

Sometimes it can be useful to hide the outliers, for example when overlaying the raw data points on top of the boxplot. Hiding the outliers can be achieved by setting `outlier.shape = NA`. Importantly, this does not remove the outliers, it only hides them, so the range calculated for the y-axis will be the same with outliers shown and outliers hidden.

`outlier.size` Default aesthetics for outliers. Set to `NULL` to inherit from the aesthetics used for the box.

In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.

Sometimes it can be useful to hide the outliers, for example when overlaying the raw data points on top of the boxplot. Hiding the outliers can be achieved by setting `outlier.shape = NA`. Importantly, this does not remove the outliers, it only hides them, so the range calculated for the y-axis will be the same with outliers shown and outliers hidden.

`outlier.stroke` Default aesthetics for outliers. Set to `NULL` to inherit from the aesthetics used for the box.

In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.

Sometimes it can be useful to hide the outliers, for example when overlaying the raw data points on top of the boxplot. Hiding the outliers can be achieved by setting `outlier.shape = NA`. Importantly, this does not remove the outliers, it only hides them, so the range calculated for the y-axis will be the same with outliers shown and outliers hidden.

`outlier.alpha` Default aesthetics for outliers. Set to `NULL` to inherit from the aesthetics used for the box.

In the unlikely event you specify both US and UK spellings of colour, the US spelling will take precedence.

Sometimes it can be useful to hide the outliers, for example when overlaying the raw data points on top of the boxplot. Hiding the outliers can be achieved by setting `outlier.shape = NA`. Importantly, this does not remove the outliers, it only hides them, so the range calculated for the y-axis will be the same with outliers shown and outliers hidden.

`notch` If `FALSE` (default) make a standard box plot. If `TRUE`, make a notched box plot. Notches are used to compare groups; if the notches of two boxes do not overlap, this suggests that the medians are significantly different.

`notchwidth` For a notched box plot, width of the notch relative to the body (defaults to `notchwidth = 0.5`).

`varwidth` If `FALSE` (default) make a standard box plot. If `TRUE`, boxes are drawn with widths proportional to the square-roots of the number of observations in the groups (possibly weighted, using the `weight` aesthetic).

`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()`.

Value

Returns an object of class `GeomQqboxplot`, (inherits from `Geom`, `ggproto`), that renders the data for the Q-Q boxplot.

Description

The Q-Q boxplot inherits its summary statistics from the boxplot. See `geom_boxplot()` for details. The Q-Q boxplot differs from the boxplot by using more informative whiskers than the regular boxplot.

The vertical position of the whiskers can be interpreted as it is in the boxplot, and the maximal vertical value is chosen as it is done in the regular boxplot. The horizontal positioning of the whiskers indicates the deviation of the data set of interest from some reference data set (specified as either a theoretical distribution or an actual data set). Taking the central vertical axis of the boxplot as being zero, deviations to the right indicate that those values are larger than the corresponding data points in the reference data set, where two data points correspond if their quantiles match. Deviations to the left indicate that the values are smaller than their corresponding data points. Consider a situation where your data set has fatter tails than the normal distribution. When the reference distribution is the normal distribution, then the whiskers below the box will be left of the central axis (the left tail values are smaller than they ought to be) and the whiskers above the box will be right of the central axis (the right tail values are larger than the ought to be).

In order to compare the data set of interest to the reference data set, they must be on the same scale. The Q-Q boxplot uses Tukey's g-h distribution to determine the appropriate scaling factor.

Much of the code here is a modification of the geom_boxplot() code.

Examples

```
p <- ggplot2::ggplot(simulated_data, ggplot2::aes(factor(group,
levels=c("normal, mean=2", "t distribution, df=32", "t distribution, df=16",
"t distribution, df=8", "t distribution, df=4")), y=y))
p + geom_qqboxplot()
p + geom_qqboxplot(reference_dist = "norm")


p + geom_qqboxplot(compdata = comparison_dataset)


# geom_qqboxplot inherits all arguments from geom_boxplot, e.g.:
p + geom_qqboxplot(notch = TRUE)
p + geom_qqboxplot(varwidth=TRUE)
p + geom_qqboxplot(ggplot2::aes(color = group)) + ggplot2::guides(color=FALSE)
```

`indicators` *World Bank indicator data for Labor Force participation rates*

Description

A dataset that contains participation rates (%) for ages 15-24, separated by gender, and measured in the years 2008, 2012, and 2017

Usage

```
indicators
```

Format

A data frame with 612 rows and 7 variables:

Country Name name of country

Country Code unique country identifier (string)

Series Name Specifies male/female

Series Code unique identifier for series

year year for data

indicator participation rate in percents

log_indicator the log of the participation rate ...

Source

https://www.worldbank.org/en/home

---

population_brain_data *Neuron population firing data*

Description

A dataset that contains populations of neurons from CA1 and LM and their firing rates for three situations: base firing rate, dot motion, and drifting gradient. Each row represents a neuron

Usage

```
population_brain_data
```

Format

A data frame with 13731 rows and 3 variables:

ecephys_structure_acronym acronym for population location

fr_type situation under which firing rate was recorded

rate the firing rate ...

Source

https://allensdk.readthedocs.io/en/latest/visual_coding_neuropixels.html

---

qqboxplot `qqboxplot` *package*

Description

Create qq-boxplots

`simulated_data` *Simulated t-distributions to show use of q-q boxplots*

Description

A dataset that contains simulated data to reproduce the simulated data figures used in our manuscript

Usage

```
simulated_data
```

Format

A data frame with 4500 rows and 2 variables:

y a value simulated from a distribution

group a string specifying the distribution from which the y value is drawn ...

Source

simulations

`spike_data` *Neuron spiking data for neural tuning orientation*

Description

A dataset that contains the number of spikes for neurons across several possible orientations of a grating

Usage

```
spike_data
```

Format

A data frame with 12800 rows and 5 variables:

orientation 1 to 8, specifies the orientation of the grating

nspikes number of spikes for a single trial of 1.28 seconds for a particular orientation

region region of the brain where the neuron is located ...

Source

https://CRCNS.org

`stat_qqboxplot` *Compute values for the Q-Q Boxplot*

Description

Compute values for the Q-Q Boxplot

Usage

```
stat_qqboxplot(
  mapping = NULL,
  data = NULL,
  geom = "qqboxplot",
  position = "dodge2",
  ...,
  coef = 1.5,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  reference_dist = "norm",
  confidence_level = 0.95,
  numboots = 500,
  qtype = 7,
  compdata = 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)`).

`geom`
: Use to override the default connection between `geom_boxplot()` and `stat_boxplot()`.

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

`...`
: 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.

`coef` Length of the whiskers as multiple of IQR. Defaults to 1.5.

`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()`.

`reference_dist` Specifies theoretical reference distribution.

`confidence_level`
Sets confidence level for deviation whisker confidence bands

`numboots` specifies the number of bootstrap draws for bootstrapped CIs needed only if compdata is not NULL

`qtype` an integer between 1 and 9 indicating which one of the quantile algorithms to use.

`compdata` specifies a data set to use as the reference distribution. If compdata is not NULL, the argument reference_dist will be ignored.

Value

Returns an object of class `StatQqboxplot`, (inherits from `Geom`, `ggproto`), that helps to render the data for `geom_qqboxplot()`.

Computed variables

`stat_qqboxplot()` provides the following variables, some of which depend on the orientation:

width width of boxplot

ymin *or* xmin lower whisker = smallest observation greater than or equal to lower hinge - 1.5 * IQR

lower *or* xlower lower hinge, 25% quantile

notchlower lower edge of notch = median - 1.58 * IQR / sqrt(n)

middle *or* xmiddle median, 50% quantile

notchupper upper edge of notch = median + 1.58 * IQR / sqrt(n)

upper *or* xupper upper hinge, 75% quantile

ymax *or* xmax upper whisker = largest observation less than or equal to upper hinge + 1.5 * IQR

Index