

Package ‘ggplot2.utils’

January 9, 2024

Type Package

Title Selected Utilities Extending 'ggplot2'

Version 0.3.1

Date 2023-12-12

Description Selected utilities, in particular 'geoms' and 'stats' functions, extending the 'ggplot2' package. This package imports functions from 'EnvStats' <[doi:10.1007/978-1-4614-8456-1](https://doi.org/10.1007/978-1-4614-8456-1)> by Millard (2013), 'ggpp' <<https://CRAN.R-project.org/package=ggpp>> by Aphalo et al. (2023) and 'ggstats' <[doi:10.5281/zenodo.10183964](https://doi.org/10.5281/zenodo.10183964)> by Larmarange (2023), and then exports them. This package also contains modified code from 'ggquicked' <<https://CRAN.R-project.org/package=ggquicked>> by Mouksassi et al. (2023) for Kaplan-Meier lines and ticks additions to plots. All functions are tested to make sure that they work reliably.

License Apache License 2.0

URL <https://insightengineering.github.io/ggplot2.utils/>

BugReports <https://github.com/insightengineering/ggplot2.utils/issues>

Depends ggplot2 (>= 3.3.0), R (>= 3.6)

Imports checkmate, EnvStats, ggpp, ggstats, survival

Suggests dplyr, lifecycle, testthat (>= 3.0.0), tibble, vdiff

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.2.3

Collate 'geom_km.R' 'geom_km_ticks.R' 'geom_table.R' 'ggproto.R' 'package.R' 'stat_km_compute.R' 'stat_km.R' 'stat_km_ticks.R' 'stat_n_text.R' 'stat_prop.R' 'table_themes.R' 'ttheme_set.R'

NeedsCompilation no

Author Daniel Sabanés Bové [aut, cre],
Samer Mouksassi [aut] (wrote original Kaplan-Meier code),
Michael Sachs [aut] (wrote original Kaplan-Meier code),
F. Hoffmann-La Roche AG [cph, fnd]

Maintainer Daniel Sabanés Bové <daniel.sabanes_bove@roche.com>

Repository CRAN

Date/Publication 2024-01-08 23:10:08 UTC

R topics documented:

ggplot2.utils-package	2
geom_km	3
geom_km_ticks	5
geom_table	7
stat_km	7
stat_km_ticks	9
stat_n_text	10
stat_prop	11
ttheme_ggdefault	11
ttheme_set	11
Index	12

ggplot2.utils-package ggplot2.utils *Package*

Description

ggplot2.utils provides simple access to utility functions extending ggplot2.

Details

Currently all of the functions are imported from other extension packages:

- ggpp: `geom_table()` and associated functions.
- EnvStats: `stat_n_text()` and associated functions.
- ggstats: `stat_prop()` and associated functions.

Author(s)

Maintainer: Daniel Sabanés Bové <daniel.sabanes_bove@roche.com>

Authors:

- Samer Mouksassi (wrote original Kaplan-Meier code)
- Michael Sachs (wrote original Kaplan-Meier code)

Other contributors:

- F. Hoffmann-La Roche AG [copyright holder, funder]

See Also

Useful links:

- <https://insightsengineering.github.io/ggplot2.utils/>
- Report bugs at <https://github.com/insightsengineering/ggplot2.utils/issues>

 geom_km

Add a Kaplan-Meier Survival Curve

Description

[Experimental] Adds the Kaplan-Meier survival curve.

Usage

```
geom_km(
  mapping = NULL,
  data = NULL,
  stat = "km",
  position = "identity",
  show.legend = NA,
  inherit.aes = TRUE,
  na.rm = TRUE,
  ...
)
```

Arguments

mapping	Set of aesthetic mappings created by <code>aes()</code> . If specified and <code>inherit.aes = TRUE</code> (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 <code>NULL</code> , the default, the data is inherited from the plot data as specified in the call to <code>ggplot()</code> . A <code>data.frame</code> , or other object, will override the plot data. All objects will be fortified to produce a data frame. See <code>fortify()</code> for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code> , and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).
stat	The statistical transformation to use on the data for this layer, either as a ggproto <code>Geom</code> subclass or as a string naming the stat stripped of the <code>stat_</code> prefix (e.g. "count" rather than "stat_count")
position	Position adjustment, either as a string naming the adjustment (e.g. "jitter" to use <code>position_jitter</code>), or the result of a call to a position adjustment function. Use the latter if you need to change the settings of the adjustment.

<code>show.legend</code>	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.
<code>inherit.aes</code>	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. <code>borders()</code> .
<code>na.rm</code>	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
<code>...</code>	Other arguments passed on to <code>layer()</code> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.

Aesthetics

`geom_km()` understands the following aesthetics (required aesthetics in bold):

- **x**: the survival/censoring times, automatically mapped by `stat_km()`.
- **y**: the survival probability estimates, automatically mapped by `stat_km()`.
- `alpha`
- `color`
- `linetype`
- `linewidth`

Author(s)

Inspired by `geom_km` written by Michael Sachs (in `ggkm`) and Samer Mouksassi (in `ggquicked`). Here we directly use `ggplot2::geom_step()` instead of the more general `ggplot2::geom_path()`.

See Also

The default stat for this geom is `stat_km()`.

Examples

```
library(ggplot2)
sex <- rbinom(250, 1, .5)
df <- data.frame(
  time = exp(rnorm(250, mean = sex)),
  status = rbinom(250, 1, .75),
  sex = sex
)
ggplot(df, aes(time = time, status = status, color = factor(sex))) +
  geom_km()
```

Description

[Experimental] Adds tickmarks at the times when there are censored observations but no events.

Usage

```
geom_km_ticks(
  mapping = NULL,
  data = NULL,
  stat = "km_ticks",
  position = "identity",
  show.legend = NA,
  inherit.aes = TRUE,
  na.rm = TRUE,
  ...
)
```

Arguments

mapping	Set of aesthetic mappings created by aes() . If specified and <code>inherit.aes = TRUE</code> (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 <code>NULL</code> , the default, the data is inherited from the plot data as specified in the call to ggplot() . A <code>data.frame</code> , 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 <code>data.frame</code> , and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).
stat	The statistical transformation to use on the data for this layer, either as a <code>ggproto</code> <code>Geom</code> subclass or as a string naming the stat stripped of the <code>stat_</code> prefix (e.g. "count" rather than "stat_count")
position	Position adjustment, either as a string naming the adjustment (e.g. "jitter" to use <code>position_jitter</code>), or the result of a call to a position adjustment function. Use the latter if you need to change the settings of the adjustment.
show.legend	logical. Should this layer be included in the legends? <code>NA</code> , the default, includes if any aesthetics are mapped. <code>FALSE</code> never includes, and <code>TRUE</code> always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes	If <code>FALSE</code> , 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() .

na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
...	Other arguments passed on to <code>layer()</code> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.

Aesthetics

`geom_km_ticks()` understands the following aesthetics (required aesthetics in bold):

- **x**: the survival/censoring times, automatically mapped by `stat_km_ticks()`.
- **y**: the survival probability estimates, automatically mapped by `stat_km_ticks()`.
- alpha
- color
- shape
- size
- stroke
- fill

Author(s)

Michael Sachs (in ggkm), Samer Mouksassi (in ggquickeda).

See Also

The default stat for this geom is `stat_km_ticks()`.

Examples

```
library(ggplot2)
sex <- rbinom(250, 1, .5)
df <- data.frame(
  time = exp(rnorm(250, mean = sex)),
  status = rbinom(250, 1, .75),
  sex = sex
)
ggplot(df, aes(time = time, status = status, color = factor(sex), group = factor(sex))) +
  geom_km() +
  geom_km_ticks(col = "black")
```

geom_table	<i>Inset tables</i>
------------	---------------------

Description**[Experimental]**

See `ggpp::geom_table()` for details.

Value

A plot layer instance.

stat_km	<i>Adds a Kaplan-Meier Estimate of Survival Statistic</i>
---------	-----------------------------------------------------------

Description

[Experimental] This stat is for computing the Kaplan-Meier survival estimate for right-censored data. It requires the aesthetic mapping time for the observation times and status which indicates the event status, either 0 for alive and 1 for dead, or 1 for alive and 2 for dead.

Usage

```
stat_km(
  mapping = NULL,
  data = NULL,
  geom = "km",
  position = "identity",
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)
```

Arguments

mapping	Set of aesthetic mappings created by <code>aes()</code> . If specified and <code>inherit.aes = TRUE</code> (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 <code>NULL</code> , the default, the data is inherited from the plot data as specified in the call to <code>ggplot()</code> . A <code>data.frame</code> , or other object, will override the plot data. All objects will be fortified to produce a data frame. See <code>fortify()</code> 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 to display the data, either as a ggproto Geom subclass or as a string naming the geom stripped of the <code>geom_</code> prefix (e.g. "point" rather than "geom_point")
position	Position adjustment, either as a string naming the adjustment (e.g. "jitter" to use <code>position_jitter</code>), or the result of a call to a position adjustment function. Use the latter if you need to change the settings of the adjustment.
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. <code>borders()</code> .
...	Other arguments passed on to <code>layer()</code> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.

Value

A `data.frame` with columns:

- `time`: time in data.
- `survival`: survival estimate at time.

Note

Logical status is not supported.

Author(s)

Michael Sachs (in `ggkm`), Samer Mouksassi (in `ggquickeda`).

Examples

```
library(ggplot2)
sex <- rbinom(250, 1, .5)
df <- data.frame(
  time = exp(rnorm(250, mean = sex)),
  status = rbinom(250, 1, .75),
  sex = sex
)
ggplot(df, aes(time = time, status = status, color = factor(sex))) +
  stat_km()
```


stat_km_ticks

Adds Tick Marks to a Kaplan-Meier Estimate of Survival Statistic

Description

[Experimental] This `stat` is for computing the location of the tick marks for the Kaplan-Meier survival estimate for right-censored data. It requires the aesthetic mapping `time` for the observation times and `status` which indicates the event status, either 0 for alive and 1 for dead, or 1 for alive and 2 for dead.

Usage

```
stat_km_ticks(
  mapping = NULL,
  data = NULL,
  geom = "km_ticks",
  position = "identity",
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)
```

Arguments

<code>mapping</code>	Set of aesthetic mappings created by <code>aes()</code> . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply <code>mapping</code> if there is no plot mapping.
<code>data</code>	The data to be displayed in this layer. There are three options: If <code>NULL</code> , the default, the data is inherited from the plot data as specified in the call to <code>ggplot()</code> . A <code>data.frame</code> , or other object, will override the plot data. All objects will be fortified to produce a data frame. See <code>fortify()</code> for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code> , and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).
<code>geom</code>	The geometric object to use to display the data, either as a ggproto <code>Geom</code> subclass or as a string naming the geom stripped of the <code>geom_</code> prefix (e.g. "point" rather than "geom_point")
<code>position</code>	Position adjustment, either as a string naming the adjustment (e.g. "jitter" to use <code>position_jitter</code>), or the result of a call to a position adjustment function. Use the latter if you need to change the settings of the adjustment.
<code>show.legend</code>	logical. Should this layer be included in the legends? <code>NA</code> , the default, includes if any aesthetics are mapped. <code>FALSE</code> never includes, and <code>TRUE</code> 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()`.

... 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`.

Value

A `data.frame` with columns:

- `time`: time in data.
- `survival`: survival estimate at time.
- `n.risk`: number of patients at risk.
- `n.censor`: number of patients censored.
- `n.event`: number of patients with event.

Note

Logical status is not supported.

Author(s)

Michael Sachs (in `ggkm`), Samer Mouksassi (in `ggquickeda`).

Examples

```
library(ggplot2)
sex <- rbinom(250, 1, .5)
df <- data.frame(
  time = exp(rnorm(250, mean = sex)),
  status = rbinom(250, 1, .75),
  sex = sex
)
ggplot(df, aes(time = time, status = status, color = factor(sex))) +
  stat_km() +
  stat_km_ticks()
```

stat_n_text

Add Text Indicating the Sample Size to a ggplot2 Plot

Description

[Experimental]

See `EnvStats::stat_n_text()` for details.

Value

A plot layer including the sample size text.

stat_prop	<i>Compute Proportions According to Custom Denominator</i>
-----------	------------------------------------------------------------

Description**[Experimental]**See `ggstats::stat_prop()` for details.**Value**

A plot layer containing the custom proportions.

ttheme_gtdefault	<i>Table themes</i>
------------------	---------------------

Description**[Experimental]**See `ggpp::ttheme_gtdefault()` for details.**Value**A list object that can be used as `ttheme` in the construction of tables with functions from package 'gridExtra'.

ttheme_set	<i>Set default table theme</i>
------------	--------------------------------

Description**[Experimental]**See `ggpp::ttheme_set()` for details.**Value**

A named list with the previous value of the option.

Note

When testing this function, we found that in contrast to the original documentation, the theme is not fixed when the plot object is constructed. Instead, the option setting affects the rendering of ready built plot objects.

Index

`aes()`, [3](#), [5](#), [7](#), [9](#)
`borders()`, [4](#), [5](#), [8](#), [10](#)
`EnvStats::stat_n_text()`, [10](#)
`fortify()`, [3](#), [5](#), [7](#), [9](#)
`geom_km`, [3](#)
`geom_km_ticks`, [5](#)
`geom_table`, [7](#)
`geom_table()`, [2](#)
`geom_table_npc` (`geom_table`), [7](#)
`ggplot()`, [3](#), [5](#), [7](#), [9](#)
`ggplot2.utils` (`ggplot2.utils`-package), [2](#)
`ggplot2.utils`-package, [2](#)
`ggplot2::geom_path()`, [4](#)
`ggplot2::geom_step()`, [4](#)
`ggpp::geom_table()`, [7](#)
`ggpp::ttheme_gtdefault()`, [11](#)
`ggpp::ttheme_set()`, [11](#)
`ggstats::stat_prop()`, [11](#)
`layer()`, [4](#), [6](#), [8](#), [10](#)
`stat_km`, [7](#)
`stat_km()`, [4](#)
`stat_km_ticks`, [9](#)
`stat_km_ticks()`, [6](#)
`stat_n_text`, [10](#)
`stat_n_text()`, [2](#)
`stat_prop`, [11](#)
`stat_prop()`, [2](#)
`StatProp` (`stat_prop`), [11](#)
`ttheme_gtbw` (`ttheme_gtdefault`), [11](#)
`ttheme_gtdark` (`ttheme_gtdefault`), [11](#)
`ttheme_gtdefault`, [11](#)
`ttheme_gtlight` (`ttheme_gtdefault`), [11](#)
`ttheme_gtminimal` (`ttheme_gtdefault`), [11](#)
`ttheme_gtplain` (`ttheme_gtdefault`), [11](#)
`ttheme_gtsimple` (`ttheme_gtdefault`), [11](#)
`ttheme_gtstripes` (`ttheme_gtdefault`), [11](#)
`ttheme_set`, [11](#)