Package ‘ggquickeda’

June 22, 2021

Title Quickly Explore Your Data Using 'ggplot2' and 'table1' Summary Tables

Version 0.2.1

Description Quickly and easily perform exploratory data analysis by uploading your data as a 'csv' file. Start generating insights using 'ggplot2' plots and 'table1' tables with descriptive stats, all using an easy-to-use point and click 'Shiny' interface.

URL `https://github.com/smouksassi/ggquickeda`

BugReports `https://github.com/smouksassi/ggquickeda/issues`

Depends R (>= 3.6.0)

Imports colourpicker, dplyr, DT, Formula, GGally (>= 2.1.0), ggbeeswarm, ggplot2 (>= 3.3.4), ggpmisc, ggrepel (>= 0.7.0), ggstance, ggpubr, gridExtra, Hmisc, markdown, methods, plotly, quantreg, rlang, scales, shiny (>= 1.0.4), shinyjs (>= 1.1), shinyjqui, stringr, survival, survminer, tidyr, table1 (>= 1.2), zoo, shinyFiles, RPostgres

Suggests knitr, rmarkdown

SystemRequirements pandoc with https support

LazyData true

VignetteBuilder knitr

RoxygenNote 7.1.1

NeedsCompilation no

Author Samer Mouksassi [aut, cre] (<`https://orcid.org/0000-0002-7152-6654`>),
Dean Attali [aut],
Benjamin Rich [aut] (provided summary stats tables tab code),
Michael Sachs [aut] (provided ggkm code)

Maintainer Samer Mouksassi <`samermouksassi@gmail.com`>

Repository CRAN

Date/Publication 2021-06-22 14:00:05 UTC

R topics documented:

`geom_km` *Add a Kaplan-Meier survival curve*

Description

Add a 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 `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`	The statistical transformation to use on the data for this layer, as a string.
`position`	Position adjustment, either as a string, or the result of a call to a position adjustment function.
`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()`.
`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 `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.

Aesthetics

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

- x The survival/censoring times. This is automatically mapped by stat_km
- y The survival probability estimates. This is automatically mapped by stat_km smallest level in sort order is assumed to be 0, with a warning
- `alpha`
- `color`
- `linetype`
- `size`

See Also

The default stat for this geom is `stat_km` see that documentation for more options to control the underlying statistical transformation.

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()
```

`geom_kmband` *Add confidence bands to a Kaplan-Meier survival curve*

Description

Add confidence bands to a Kaplan-Meier survival curve

Usage

```
geom_kmband(
  mapping = NULL,
  data = NULL,
  stat = "kmband",
  position = "identity",
  show.legend = NA,
  inherit.aes = TRUE,
  na.rm = 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`	The statistical transformation to use on the data for this layer, as a string.
`position`	Position adjustment, either as a string, or the result of a call to a position adjustment function.
`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()`.
`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 `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.

Aesthetics

`geom_kmband` understands the following aesthetics (required aesthetics are in bold):

- `x` The survival/censoring times. This is automatically mapped by stat_kmband
- `y` The survival probability estimates. This is automatically mapped by stat_kmband smallest level in sort order is assumed to be 0, with a warning
- `alpha`
- `color`
- `linetype`
- `size`

See Also

The default stat for this geom is `stat_kmband` see that documentation for more options to control the underlying statistical transformation.

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), fill =factor(sex))) +
 geom_km() + geom_kmband()
```

`geom_kmticks` *Add tick marks to a Kaplan-Meier survival curve*

Description

Adds tickmarks at the times when there are censored observations but no events

Usage

```
geom_kmticks(
  mapping = NULL,
  data = NULL,
  stat = "kmticks",
  position = "identity",
  show.legend = NA,
  inherit.aes = TRUE,
  na.rm = 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`	The statistical transformation to use on the data for this layer, as a string.
`position`	Position adjustment, either as a string, or the result of a call to a position adjustment function.
`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()`.
`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 `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.

Aesthetics

`geom_kmticks` understands the following aesthetics (required aesthetics are in bold):

- x The survival/censoring times. This is automatically mapped by stat_km
- y The survival probability estimates. This is automatically mapped by stat_km smallest level in sort order is assumed to be 0, with a warning
- `alpha`
- `color`
- `linetype`
- `size`

See Also

The default stat for this geom is `stat_kmticks` see that documentation for more options to control the underlying statistical transformation.

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_kmticks(col="black")
```

`run_ggquickeda` *Run the ggquickeda application*

Description

Run the `ggquickeda` application.

Usage

```
run_ggquickeda(data = NULL)
```

Arguments

`data`	The initial data.frame to load into the application.

Examples

```
if (interactive()) {
  run_ggquickeda()
}
```

`sample_data` *Simulated Pharmacokinetic Concentration Data*

Description

A dataset containing concentration-time data with the given dose and some subject characteristics to help in the app exploration.

Usage

```
sample_data
```

Format

A data frame with 600 rows and 10 variables

ID Subject Identifier, an integer from 1 to 150

Time Time of dose given or drug sample measured, in hours

Amt dose given at the corresponding Time, in milligrams

Conc drug concentrations in the plasma sample, in mg/L

Age age of the subject, in years

Weight weight of the subject, in kg

Gender Sex of the subject, a factor with Female and Male levels

Race Race of the subject, a factor with Asian, Black, Caucasian, Hispanic and Other levels

Dose dose group of the subject, in milligrams

AGECAT age category of the subject, a variable cutting Age into two values 0/1

Source

"sd_oral_richpk" from ’PKPDmisc’ R package with an additional AGECAT variable

---

`stat_km` *Adds a Kaplan Meier Estimate of Survival*

Description

Adds a Kaplan Meier Estimate of Survival

Usage

```
stat_km(
  mapping = NULL,
  data = NULL,
  geom = "km",
  position = "identity",
  show.legend = NA,
  inherit.aes = TRUE,
  trans = scales::identity_trans(),
  firstx = 0,
  firsty = 1,
  type = "kaplan-meier",
  start.time = 0,
  ...
)
```

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.
`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()`.
`trans`	Transformation to apply to the survival probabilities. Defaults to "identity". Other options include "event", "cumhaz", "cloglog", or define your own using trans_new.
`firstx, firsty`	the starting point for the survival curves. By default, the plot program obeys tradition by having the plot start at (0,1).
`type`	an older argument that combined stype and ctype, now deprecated. Legal values were "kaplan-meier" which is equivalent to stype=1, ctype=1, "fleming-harrington" which is equivalent to stype=2, ctype=1, and "fh2" which is equivalent to stype=2, ctype=2.
`start.time`	numeric value specifying a time to start calculating survival information. The resulting curve is the survival conditional on surviving to start.time.
`...`	Other arguments passed to survfit.formula

Details

This stat is for computing the confidence intervals for the Kaplan-Meier survival estimate for right-censored data. It requires the aesthetic mapping `x` for the observation times and `status` which indicates the event status, 0=alive, 1=dead or 1/2 (2=death). Logical status is not supported.

Value

a data.frame with additional columns:

`x`	x in data
`y`	Kaplan-Meier Survival Estimate at x

Aesthetics

`stat_km` understands the following aesthetics (required aesthetics are in bold):

- `time` The survival times
- `status` The censoring indicator, see Surv for more information.
- `alpha`
- `color`
- `linetype`
- `size`

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()

## Examples illustrating the options passed to survfit.formula

p1 <- ggplot(df, aes(time = time, status = status))
p1 + stat_km()
p1 + stat_km(trans = "cumhaz")
# for cloglog plots also log transform the time axis
p1 + stat_km(trans = "cloglog") + scale_x_log10()
p1 + stat_km(type = "fleming-harrington")
p1 + stat_km(start.time = 5)
```

`stat_kmband` *Adds confidence bands to a Kaplan Meier Estimate of Survival*

Description

Adds confidence bands to a Kaplan Meier Estimate of Survival

Usage

```
stat_kmband(
  mapping = NULL,
  data = NULL,
  geom = "kmband",
  position = "identity",
  show.legend = NA,
  inherit.aes = TRUE,
  trans = "identity",
  firstx = 0,
```

```
    firsty = 1,
    type = "kaplan-meier",
    error = "greenwood",
    conf.type = "log",
    conf.lower = "usual",
    start.time = 0,
    conf.int = 0.95,
    ...
)
```

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.
`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()`.
`trans`	Transformation to apply to the survival probabilities. Defaults to "identity". Other options include "event", "cumhaz", "cloglog", or define your own using trans_new.
`firstx, firsty`	the starting point for the survival curves. By default, the plot program obeys tradition by having the plot start at (0,1).
`type`	an older argument that combined stype and ctype, now deprecated. Legal values were "kaplan-meier" which is equivalent to stype=1, ctype=1, "fleming-harrington" which is equivalent to stype=2, ctype=1, and "fh2" which is equivalent to stype=2, ctype=2.
`error`	either the string "greenwood" for the Greenwood formula or "tsiatis" for the Tsiatis formula, (only the first character is necessary). The default is "greenwood".
`conf.type`	One of "none", "plain", "log" (the default), "log-log" or "logit".

`conf.lower`	a character string to specify modified lower limits to the curve, the upper limit remains unchanged. Possible values are "usual" (unmodified), "peto", and "modified". The modified lower limit is based on an "effective n" argument. The confidence bands will agree with the usual calculation at each death time, but unlike the usual bands the confidence interval becomes wider at each censored observation. The extra width is obtained by multiplying the usual variance by a factor m/n, where n is the number currently at risk and m is the number at risk at the last death time. (The bands thus agree with the un-modified bands at each death time.) This is especially useful for survival curves with a long flat tail. The Peto lower limit is based on the same "effective n" argument as the modified limit, but also replaces the usual Greenwood variance term with a simple approximation. It is known to be conservative.
`start.time`	numeric value specifying a time to start calculating survival information. The resulting curve is the survival conditional on surviving to start.time.
`conf.int`	the level for a two-sided confidence interval on the survival curve(s). Default is 0.95.
`...`	Other arguments passed to survfit.formula

Details

This stat is for computing the confidence intervals for the Kaplan-Meier survival estimate for right-censored data. It requires the aesthetic mapping `x` for the observation times and `status` which indicates the event status, 0=alive, 1=dead or 1/2 (2=death). Logical status is not supported.

Value

a data.frame with additional columns:

`x`	x in data
`ymin`	Lower confidence limit of KM curve
`ymax`	Upper confidence limit of KM curve

Aesthetics

`stat_kmband` understands the following aesthetics (required aesthetics are in bold):

- `time` The survival times
- `status` The censoring indicator, see Surv for more information.
- `alpha`
- `color`
- `linetype`
- `size`

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()

## Examples illustrating the options passed to survfit.formula

p1 <- ggplot(df, aes(time = time, status = status))
p1 + stat_km() + stat_kmband(conf.int = .99)
p1 + stat_kmband(error = "greenwood",fill="red",alpha=0.2) +
 stat_kmband(error = "tsiatis",fill="blue",alpha=0.2)+ stat_km()
p1 + stat_km() + stat_kmband(conf.type = "log-log")+ stat_kmband(conf.type = "log")
```

`stat_kmticks` *Adds tick marks to a Kaplan Meier Estimate of Survival*

Description

Adds tick marks to a Kaplan Meier Estimate of Survival

Usage

```
stat_kmticks(
  mapping = NULL,
  data = NULL,
  geom = "kmticks",
  position = "identity",
  show.legend = NA,
  inherit.aes = TRUE,
  trans,
  ...
)
```

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.
`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()`.
`trans`	Transformation to apply to the survival probabilities. Defaults to "identity". Other options include "event", "cumhaz", "cloglog", or define your own using trans_new.
`...`	Other arguments passed to survfit.formula

Details

This stat is for computing the tick marks for a Kaplan-Meier survival estimate for right-censored data. The tick marks will appear at each censoring time which is also not a death time, which is the default for plot.survfit. It requires the aesthetic mapping `x` for the observation times and `status` which indicates the event status, normally 0=alive, 1=dead. Other choices are TRUE/FALSE (TRUE = death) or 1/2 (2=death).

Value

a data.frame with additional columns:

x	x in data
y	Kaplan-Meier Survival Estimate at x

Aesthetics

`stat_kmticks` understands the following aesthetics (required aesthetics are in bold):

- `time` The survival times
- `status` The censoring indicator, see Surv for more information.
- `alpha`
- `color`
- `linetype`
- `size`

See Also

`stat_km`; `stat_kmband`

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_kmticks()
```

Index