Package ‘utile.visuals’

February 14, 2022

Title Create Visuals for Publication
Version 0.3.1
Description A small set of functions to aid in the production of visuals in ggplot2. Includes minimalist themes with transparent backgrounds and tools for building survival curves with risk tables.
License LGPL (>= 2)
URL https://github.com/efinite/utile.visuals
BugReports https://github.com/efinite/utile.visuals/issues
Encoding UTF-8
Depends R (>= 3.4.0)
Imports dplyr, ggplot2, gridExtra, purrr
Suggests survival, broom, grid
RoxygenNote 7.1.2
NeedsCompilation no
Author Eric Finnesgard [aut, cre]
Maintainer Eric Finnesgard <efinite@outlook.com>
Repository CRAN
Date/Publication 2022-02-14 17:40:02 UTC

R topics documented:

append_table ................................................................. 2
ggeom_stepconfint .......................................................... 3
ggrisktable ................................................................. 5
theme_basic ................................................................. 6
theme_risk ................................................................. 6

Index 8
Append a ggplot2 table to the bottom of a ggplot2 plot

Description

Aligns axes and combines a ggplot2 plot and table into a single plot. Can handle legends.

Usage

```r
append_table(
  plot = NULL,
  table = NULL,
  plot.height = 1,
  table.height = 0.1,
  plot.width = 1,
  extract.legend = TRUE,
  legend.width = 0.2,
  legend.offset = -15
)
```

Arguments

- `plot`: Required. ggplot2::ggplot() object. If a legend is present, it will be extracted.
- `table`: Required. ggplot2::ggplot object. If a legend is present, it will be removed and ignored.
- `extract.legend`: Optional. Logical. Indicates whether to extract the legend from the plot and reinsert it adjacent to the final combined plot. May be undesired if legend already embedded within the plot area. Defaults to TRUE.
- `legend.width`: Optional. Numeric. Width of legend relative to plot. Ignored if no legend present in plot or `extract.legend`=FALSE. Defaults to 0.2.

Value

A ggplot2 tableGrob object. Use grid::grid.draw() to open in RStudio viewer. Works with ggplot2::ggsave() out of the box.

Note

To ensure proper alignment, double check that both plots use the same scale and breaks!
Examples

```r
library(survival)
library(ggplot2)
library(broom) # tidy() model data
library(grid) # grid.draw() finished plot

# Data with group names specified
data_diabetic <- diabetic
data_diabetic$trt <- as.factor(data_diabetic$trt)
levels(data_diabetic$trt) <- c('None', 'Laser')

# Survival Model
fit <- survfit(Surv(time, status) ~ trt, data = data_diabetic)

# Kaplan Meier (KM) Plot
plot_km <- ggplot(
data = tidy(fit),
mapping = aes(x = time, y = estimate)
) +
  geom_step(aes(color = strata)) +
  geom_stepconfint(aes(ymin = conf.low, ymax = conf.high, fill = strata), alpha = 0.3) +
  coord_cartesian(c(0, 50)) + # Note scale set here!
  scale_x_continuous(expand = c(0.02,0)) +
  labs(x = 'Time', y = 'Freedom From Event') +
  scale_color_manual(
    values = c('#D83641', '#1A45A7'),
    name = 'Treatment',
    labels = c('Laser', 'None'),
    aesthetics = c('colour', 'fill')) +
  theme_basic()

# Risk Table
tbl_risk <- ggrisktable(fit, c(0, 10, 20, 30, 40, 50)) +
  coord_cartesian(c(0, 50)) +
  scale_x_continuous(expand = c(0.02,0)) +
  theme_risk()

# Combine KM plot and risk table
plot_cmbd <- append_table(
  plot = plot_km,
  table = tbl_risk
)

# Draw in RStudio viewer
grid.newpage()
grid.draw(plot_cmbd)
```
Description

Produces a step function confidence interval for survival curves. Essentially the geom_step() for
confidence intervals which ggplot2 does not provide.

Usage

geom_stepconfint(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  na.rm = FALSE,
  ...
)

Arguments

mapping Aesthetic mappings with aes() function. Like geom_ribbon(), you must provide
columns for x, ymin (lower limit), ymax (upper limit).
data The data to be displayed in this layer. Can inherit from ggplot parent.
stat The statistical transformation to use on the data for this layer, as a string. De-
faults to 'identity'.
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.
... Optional. Any other ggplot geom_ribbon() arguments.

Note

Adapted from the survminer package <https://github.com/kassambara/survminer>.

Examples

library(survival)
library(broom)
library(ggplot2)

fit <- survfit(Surv(time, status) ~ trt, data = diabetic)
fit <- survfit0(fit) # connect origin

ggplot(
  data = tidy(fit),
  mapping = aes(x = time, y = estimate)
) +
  geom_step(aes(color = strata)) +
  geom_stepconfint(aes(ymin = conf.low, ymax = conf.high, fill = strata), alpha = 0.3) +
  coord_cartesian(c(0, 50)) +
  scale_x_continuous(expand = c(0.02, 0)) +
ggrisktable

Create a ggplot2 table showing the number at risk

Description
A simple wrapper function which calculates the numbers at risk for a survival model and a given set of time points then creates a ggplot2 table with them.

Usage

```r
ggrisktable(
  fit = NULL,
  times = NULL,
  text.color = "black",
  strata.order = NULL
)
```

Arguments

- **fit**: Required. `survival::survfit()` object.
- **times**: Required. Numeric. One or more time points to calculate the number at risk for.
- **text.color**: Optional. Character. Color of text within table. Defaults to 'black'.
- **strata.order**: Optional. Character. Ordered names of strata factor levels.

Value
An unformatted ggplot2 table showing the number at risk.

Examples

```r
library(survival)

fit <- survfit(Surv(time, status) ~ trt, data = diabetic)

# Use ggrisktable function

ggrisktable(
  fit = fit,
  times = c(0, 10, 20, 30, 40, 50),
  strata.order = c('0', '1')
) + theme_risk()
```
### theme_basic

**Minimalist theme for ggplot2**

**Description**

A ggplot2 theme which removes most background elements.

**Usage**

```r
theme_basic(
  base_size = 12,
  base_family = "",
  base_color = "black",
  base_line_size = base_size/14,
  base_rect_size = base_size/14
)
```

**Arguments**

- **base_family** Optional. Numeric. Base font family. Defaults to ?.
- **base_color** Optional. Character. Base color for lines and text. Defaults to black.

**Note**

Recommend exporting as PNG or TIFF to preserve background transparency.

### theme_risk

**Minimalist risk table theme for ggplot2**

**Description**

A ggplot2 theme which removes most background elements.

**Usage**

```r
theme_risk(
  base_size = 12,
  base_family = "",
  base_color = "black",
  base_line_size = base_size/14,
  base_rect_size = base_size/14
)
```
Arguments

- `base_color` Optional. Character. Base color for lines and text. Defaults to black.

Note

Recommend exporting as PNG or TIFF to preserve background transparency.
Index

append_table, 2

geom_stepconfint, 3
ggrisktable, 5

theme_basic, 6
theme_risk, 6