Package ‘utile.visuals’

Title Create Visuals for Publication
Version 0.2.5
Description A small set of functions for making visuals for publication in ggplot2. Includes minimal-ist themes with transparent backgrounds and a suite of tools for building Kaplan-Meier 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
LazyData TRUE
Depends R (>= 3.4.0)
Imports dplyr, ggplot2, gridExtra, purrr
Suggests survival, broom, grid
RoxygenNote 7.1.0
NeedsCompilation no
Author Eric Finnesgard [aut, cre]
Maintainer Eric Finnesgard <efinite@outlook.com>
Repository CRAN
Date/Publication 2020-05-01 22:40:02 UTC

R topics documented:

append_table ................................................................. 2
connect_origin ............................................................. 3
geom_stepconfint .......................................................... 4
ggskrishtable ............................................................... 6
theme_black .................................................................. 7
theme_risk_black ............................................................ 7
theme_risk_white ............................................................ 8
theme_white ................................................................. 9

Index 10
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.
- **plot.height**: Optional. Numeric. Height of plot relative to table. Defaults to 1.
- **table.height**: Optional. Numeric. Height of table relative to plot. Defaults to 0.1.
- **plot.width**: Optional. Numeric. Width of plot relative to legend. Ignored if no legend present in plot. Defaults to 1.
- **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 0.2.
- **legend.offset**: Optional. Numeric. Vertical offset of legend. Used to raise or lower. Ignored if no legend present in plot or 'extract.legend'=FALSE. Defaults to -15.

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_black()

# 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_black()

# 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)
```

---

**connect_origin**

*Connect tidy'd survival::survfit data to the origin of a plot*
Description

Occasionally when tidy’d survfit data is graphed in ggplot2::geom_step(), the KM curve will not connect with the origin of the plot. This tool appends data connecting the lines to the origin.

Usage

connect_origin(data)

Arguments

data  Required. tibble::tibble() object. survival::survfit data that has been tidy’d with broom::tidy().

Value

A tibble containing the original data with appended points that connect the curve with to the plot origin.

Note

Adapted from the survminer package created by Alboukadel Kassambara.

Examples

library(survival)
library(broom)  # tidy() model data

# 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)
fit_data <- tidy(fit)
connect_origin(fit_data)

geom_stepconfint  Step function confidence intervals for ggplot2

Description

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

```r
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. Defaults to `"identity"`.
- **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.
- **...**: Optional. All the other miscellaneous `ggplot geom_ribbon()` arguments.

Note

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

Examples

```r
library(survival)
library(broom)
library(ggplot2)

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

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)) +
  labs(x = "Var Time", y = "Freedom From Event") +
  scale_color_manual(
    values = c("#D83641", "#1A45A7"),
    name = "Treatment",
    labels = c("None", "Laser"),
    aesthetics = c("colour", "fill")) +
```
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

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

library(survival)

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

ggrisktable(
  fit = fit,
  times = c(0, 10, 20, 30, 40, 50),
  strata.order = c('0', '1')
) + theme_risk_black()
**theme_black**  
*Minimalist black theme for ggplot2*

**Description**

A ggplot2 theme which removes most background elements and makes all text/lines black.

**Usage**

```r
theme_black(  
  base_size = 11,
  base_family = "",
  base_line_size = base_size/22,
  base_rect_size = base_size/22
)
```

**Arguments**


**Note**

Recommend exporting as PNG or TIFF to preserve background transparency.

---

**theme_risk_black**  
*Minimalist black risk table theme for ggplot2*

**Description**

Ggplot2 theme which removes most background elements and makes all text/lines black.

**Usage**

```r
theme_risk_black(  
  base_size = 10,
  base_family = "",
  base_line_size = base_size/22,
  base_rect_size = base_size/22
)
```
Arguments

base_family  Optional. Numeric. Base font family. Defaults to ?.

Note

Recommend exporting as PNG or TIFF to preserve background transparency.

theme_risk_white  Minimalist white risk table theme for ggplot2

Description

Ggplot2 theme which removes most background elements and makes all text/lines white.

Usage

theme_risk_white(
  base_size = 10,
  base_family = "",
  base_line_size = base_size/22,
  base_rect_size = base_size/22
)

Arguments

base_family  Optional. Numeric. Base font family. Defaults to ?.

Note

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

Minimalist white theme for ggplot2

Description

A ggplot2 theme which removes most background elements and makes all text/lines white.

Usage

theme_white(
  base_size = 11,
  base_family = "",
  base_line_size = base_size/22,
  base_rect_size = base_size/22
)

Arguments

base_family Optional. Numeric. Base font family. Defaults to ?.

Note

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

append_table, 2
connect_origin, 3
geom_stepconfint, 4
ggrisktable, 6
theme_black, 7
theme_risk_black, 7
theme_risk_white, 8
theme_white, 9