Package ‘SubgrPlots’

January 29, 2020

Type Package

Title Graphical Displays for Subgroup Analysis in Clinical Trials

Version 0.1.3

Description Provides functions for obtaining a variety of graphical displays that may be useful in the subgroup analysis setting. An example with a prostate cancer dataset is provided. The graphical techniques considered include level plots, mosaic plots, contour plots, bar charts, Venn diagrams, tree plots, forest plots, Galbraith plots, L'Abbé plots, the subpopulation treatment effect pattern plot, alluvial plots, circle plots and UpSet plots.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 2.10), graphics

Imports alluvial, circlize, colorspace, diagram, dplyr, ggplot2 (>= 2.1.1), ggrepel, grid, gridBase, gridExtra, methods, plyr, polyclip, scales, shape, sp, survRM2, survival, UpSetR, VennDiagram

Suggests knitr, rmarkdown, rgeos

VignetteBuilder knitr, rmarkdown

RoxygenNote 7.0.2

NeedsCompilation no

Author Nicolas Ballarini [aut, cre],
Yi-Da Chiu [aut],
Jake Conway [ctb],
Nils Gehlenborg [ctb],
Michal Bojanowski [ctb],
Robin Edwards [ctb]

Maintainer Nicolas Ballarini <`nicoballarini@gmail.com`>

Repository CRAN

Date/Publication 2020-01-29 05:40:06 UTC

R topics documented:

`ggplot_radial` *Radial (Galbraith) plot for subgroup effect size*

Description

This function produces a modified Galbraith's radial plot using ggplot. It shows the treatment effect size of subgroups defined by the categories of covariates. The x-axis represents the reciprocal of the standard error of subgroup treatment effect estimates.The y-axis means standardized effect size difference (the difference between subgroup effect the full popultion effect is divided by the standard error of the estimator for the overall population effect. Points here are for subgroups. The grey region indicates whether subgroup effects are homogeneous to the full population effect or not. The two arcs on the right side show subgroup treatment effects in the original scale, where the red spots are the projection of points from the origin on the left side. Note that the vertical range of display can be changed by setting different values on the associated input argument. In addition, the function uses log odd ratio and log hazard ratio for displaying subgroup effect sizes in binary and survival data, respectively. A ggplot object is returned, where further modifications can be added using the theme() function.

Usage

```
ggplot_radial(
  dat,
  covari.sel,
  trt.sel,
  resp.sel,
  outcome.type,
  range.v = NULL,
  font.size = 4,
  title = NULL,
  lab.xy = "default",
  ticks.length = 0.05
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of the two covariates
`trt.sel`	a covariate index specifying the treatment code
`resp.sel`	a covariate index specifying the response variable
`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary" or "survival".
`range.v`	a vector specifying the vertical range of graphical display.
`font.size`	a number specifying the font size for the subgroup labels near points
`title`	a string specifying the main title.
`lab.xy`	a list of two strings specifying the labels of the x and y axes.
`ticks.length`	a number from 0 to 1 to specify the length of the red ticks indicating the treatment effect in the radial axis. A value of 1 will draw a complete line from the (0,0) point to the axis.

Examples

```
library(dplyr)

# Load the data to be used
data(prca)
dat <- prca
dat %>%
  mutate(bm = factor(ifelse(bm == 0 , "No", "Yes")),
         hx = factor(ifelse(hx == 0 , "No", "Yes")))-> dat

ggplot_radial(dat,
```

```
            covari.sel = c(4, 5, 6, 7),
            trt.sel = 3,
            resp.sel = c(1, 2),
            outcome.type = "survival",
            range.v = c(-7, 6),
            font.size = 4)
```

`ggplot_radial2` *Modified Radial (Galbraith) plot for subgroup effect size*

Description

This function produces a modified Galbraith's radial plot using ggplot. It shows the treatment effect size of subgroups defined by the categories of covariates. The x-axis represents the reciprocal of the standard error of the difference between the subgroup treatment effect estimates and the overall treatment effect estimate. The y-axis is the standardized effect size difference (the difference between subgroup effect the full population effect is divided by its standard error). Points here are for subgroups. The grey region indicates whether subgroup effects are homogeneous to the full population effect or not. The two arcs on the right side show subgroup treatment effects in the original scale, where the red spots are the projection of points from the origin on the left side. Note that the vertical range of display can be changed by setting different values on the associated input argument. In addition, the function uses log odd ratio and log hazard ratio for displaying subgroup effect sizes in binary and survival data, respectively. A ggplot object is returned, where further modifications can be added using the theme() function.

Usage

```
ggplot_radial2(
  dat,
  covari.sel,
  trt.sel,
  resp.sel,
  outcome.type,
  range.v = NULL,
  font.size = 4,
  title = NULL,
  lab.xy = "default",
  ticks.length = 0.05
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of the two covariates
`trt.sel`	a covariate index specifying the treatment code
`resp.sel`	a covariate index specifying the response variable

`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary" or "survival".
`range.v`	a vector specifying the vertical range of graphical display.
`font.size`	a number specifying the font size for the subgroup labels near points
`title`	a string specifying the main title.
`lab.xy`	a list of two strings specifying the labels of the x and y axes.
`ticks.length`	a number from 0 to 1 to specify the length of the red ticks indicating the treatment effect in the radial axis. A value of 1 will draw a complete line from the (0,0) point to the axis.

Examples

```
library(dplyr)

# Load the data to be used
data(prca)
dat <- prca
dat %>%
  mutate(bm = factor(ifelse(bm == 0 , "No", "Yes")),
         hx = factor(ifelse(hx == 0 , "No", "Yes")))-> dat

ggplot_radial2(dat,
            covari.sel = c(4, 5, 6, 7),
            trt.sel = 3,
            resp.sel = c(1, 2),
            outcome.type = "survival",
            range.v = c(-7, 6),
            font.size = 4)
```

`ggplot_stepp` *STEPP for subgroup effect size*

Description

This function produces a plot of using the approach "Subpopulation Treatment Effect Pattern Plot". It shows the treatment effect size of subgroups, where subgruops are defined by certain ranges of a continuous covariate; each subgroup has a sample size close to a pre-specified value (N2) and any neighboring subgroups have an overlap size near another pre-specified value (N1). The plot shows the 95 a horizontal line). The y-coordinate of a point indicates the effect size within the corresponding subgroup; the x-coordinate shows the lower bound of the range which defines the subgroup. If part of the horizontal line is out of the simultaneous C.I., it may reveal hetergeneity across subgroup effects with repective to the overall effect. In addition, one can control the width of the C.I. by controlling Type I error rate in one of the function arguments. Note that the function uses log odd ratio and log hazard ratio for displaying subgroup effect sizes in binary and survival data, respectively. The actual subgroup sample sizes over the covariate are shown on the console window as well.

Usage

```
ggplot_stepp(
  dat,
  covari.sel,
  trt.sel,
  resp.sel,
  outcome.type,
  setup.ss,
  alpha,
  title = NULL,
  lab.y = NULL,
  subtitle = NULL
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of the two covariates
`trt.sel`	a variate index specifying the treatment code
`resp.sel`	a variate index specifying the response variable
`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary" or "survival".
`setup.ss`	a vector specifying the approximate overlap size (N2) and subgroup sample size (N1).
`alpha`	the type I error rate
`title`	a string specifying the main title.
`lab.y`	a string specifying the labels of the y-axis.
`subtitle`	strings specifying the subtitle

Details

Contrary to `plot_stepp`, `ggplot_stepp` implements the proper x-axis. This function uses the ggplot2 package to draw the actual plot. To control font sizes, use `theme` with the resulting object.

See Also

`plot_stepp`

Examples

```
# Load the data to be used
data(prca)
dat = prca

## 9. stepp Plot -----------------------------------------------------------
lab.y.title = paste("Treatment effect size (log-hazard ratio)");
setup.ss = c(30,40)
```

```
sub.title = paste0("(Subgroup sample sizes are set to ", setup.ss[2],
                   "; overlap sizes are set to ", setup.ss[1], ")" )
ggplot_stepp(dat,
           covari.sel = 8,
           trt.sel = 3,
           resp.sel = c(1, 2),
           outcome.type = "survival",
           setup.ss = c(30,40),
           alpha = 0.05,
           title = NULL,
           lab.y = lab.y.title,
           subtitle = sub.title)
```

`plot_alluvial` *Alluvial plot*

Description

this function produces an alluvial diagram. This function is a copy of alluvial::alluvial but a 'rotate' option is added to rotate the labels.

Usage

```
plot_alluvial(
  ...,
  freq,
  col = "gray",
  border = 0,
  layer,
  hide = FALSE,
  alpha = 0.5,
  gap.width = 0.05,
  xw = 0.1,
  cw = 0.1,
  blocks = TRUE,
  ordering = NULL,
  axis_labels = NULL,
  cex = par("cex"),
  cex.axis = par("cex.axis"),
  rotate = 0,
  las = 1,
  bottom.mar = 2
)
```

Arguments

`...`	vectors or data frames, all for the same number of observations
`freq`	numeric, vector of frequencies of the same length as the number of observations
`col`	vector of colors of the stripes
`border`	vector of border colors for the stripes
`layer`	numeric, order of drawing of the stripes
`hide`	logical, should particular stripe be plotted
`alpha`	numeric, vector of transparency of the stripes
`gap.width`	numeric, relative width of inter-category gaps
`xw`	numeric, the distance from the set axis to the control points of the xspline
`cw`	numeric, width of the category axis
`blocks`	logical, whether to use blocks to tie the flows together at each category, versus contiguous ribbons (also admits character value "bookends")
`ordering`	list of numeric vectors allowing to reorder the alluvia on each axis separately, see Examples
`axis_labels`	character, labels of the axes, defaults to variable names in the data
`cex`	numeric, scaling of fonts of category labels and axis labels respectively. See `par`.
`cex.axis`	cex value to be passed to the axis. See help(par)
`rotate`	angle to rotate the labels. This argument is passed as an `srt` argument
`las`	direction of the labels of the covariates. 1 is horizontal, 2 is vertical.
`bottom.mar`	bottom margin to be passed as a first argument to `mar()`

Examples

```
library(dplyr)

# Alluvial plot
# Load the data to be used
data(prca)
dat <- prca
dat$trt = dat$rx
dat %>%
  dplyr::select(trt, bm, hx, pf) %>%
  dplyr::group_by(trt, bm, hx, pf) %>%
  dplyr::summarise(Freq = n()) -> alldat
alldat %>%
  ungroup() %>%
  mutate(trt = ifelse(trt == 0 , "Control", "Treatment"),
         bm = ifelse(bm == 0 , "No", "Yes"),
         hx = ifelse(hx == 0 , "No", "Yes"),
         pf = ifelse(pf == 0 , "No", "Yes"))-> alldat

plot_alluvial(alldat[,c(1,3,2,4)], freq = alldat$Freq,
              xw=0.2,cw = 0.12,cex = 1,
```

```
              alpha  = 0.8,
              col=ifelse(alldat$trt == "Treatment","#1f78b4", "#a6cee3"),
              layer = alldat$trt  == 1, rotate = 90)


# Alluvial plot using survival rate at 24 months
data(prca)
dat <- prca
dat %>%
  mutate(survival = factor(ifelse(survtime > 24 , "Yes", "No"), levels = c("No", "Yes")),
         trt = rx) -> dat
dat %>%
  dplyr::select(trt, bm, hx, pf, survival) %>%
  dplyr::group_by(trt, bm, hx, pf, survival) %>%
  dplyr::summarise(Freq = n()) -> alldat
alldat %>%
  ungroup() %>%
  mutate(trt = ifelse(trt == 0 , "Control", "Treatment"),
         bm = ifelse(bm == 0 , "No", "Yes"),
         hx = ifelse(hx == 0 , "No", "Yes")) -> alldat

plot_alluvial(alldat[,c(5,1,3,2,4)], freq = alldat$Freq,
              xw=0.2,cw = 0.12,cex = 1,
              alpha  = 0.8,
              col=ifelse(alldat$survival  == "Yes",
                         ifelse(alldat$trt  == "Treatment","#80b1d3","#d5e2eb"),
                         ifelse(alldat$trt  == "Treatment","#faa8d2","#fbe0ee")),
              layer = alldat$trt  == 1, rotate = 90, las = 2, bottom.mar = 5)
```

`plot_barchart` *Barchart for subgroup effect size*

Description

This function produces a bar chart showing the treatment effect size of pairwise subgroups defined by the categories of two covariates. Also, it prints out the minimum and maximum of the treatment effect size on the console. Note that each bar has a width which is proportional to the ratio of the corresponding sample size to the full size. In addition, the function uses log odd ratio and log hazard ratio for displaying subgroup effect sizes in binary and survival data, respectively.

Usage

```
plot_barchart(
  dat,
  covari.sel,
  trt.sel,
  resp.sel,
  outcome.type,
```

```
  font.size = c(15, 12, 10, 0.6),
  title = NULL,
  lab.y = NULL,
  effect = "RMST",
  time = NULL,
  decimals = 0
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of the two covariates
`trt.sel`	a covariate index specifying the treatment code
`resp.sel`	a covariate index specifying the response variable
`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary" or "survival".
`font.size`	a vector specifying the size of labels and text; the first element is for the main title; the second element is for the covariates labels and the y-axis label; the third is for the category labels; the forth is for the unit label of the y axis.
`title`	a string specifying the main title.
`lab.y`	a string specifying the y-axis label.
`effect`	either "HR" or "RMST". only when outcome.type = "survival"
`time`	time for calculating the RMST
`decimals`	decimal places for the axis

Examples

```
# # Load the data to be used
data(prca)
dat <- prca
levels(dat$age_group) = c("Young","Middle-aged","Old")
levels(dat$weight_group) = c("Low","Mid","High")
names(dat)[c(14,15)] = c("Age", "Weight")

## 4. Bar chart -----------------------------------------------------------
plot_barchart(dat,
              covari.sel = c(14,15),
              trt.sel = 3,
              resp.sel = c(1, 2),
              outcome.type = "survival",
              font.size = c(14, 12, 14, 0.75), time = 50,
              lab.y = "Treatment effect size (RMST difference)")
```

`plot_circle` *Circular plot using circlize package*

Description

This function produces a circular plot for subgroup analysis using the circlize package. Overlaps across all variable levels are displayed in a single diagram.

Usage

```
plot_circle(
  dat,
  covari.sel,
  trt.sel,
  resp.sel,
  outcome.type,
  range.v = NULL,
  adj.ann.subgrp = 4,
  range.strip = c(-3, 3),
  n.brk = 31,
  n.brk.axis = NULL,
  font.size = c(1, 1, 0.85, 0.85, 1),
  title = NULL,
  lab.xy = NULL,
  strip = "Treatment effect size",
  effect = "HR",
  equal.width = TRUE,
  show.KM = FALSE,
  show.effect = TRUE,
  conf.int = TRUE,
  show.overall = TRUE,
  palette = "divergent",
  col.power = 0.5
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of the two covariates
`trt.sel`	a covariate index specifying the treatment code
`resp.sel`	a covariate index specifying the response variable
`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary" or "survival".
`range.v`	a vector specifying the vertical range of graphical display.
`adj.ann.subgrp`	a parameter adjusting the distance between a point and its corresponding subgroup label. The smaller the value is, the larger the distance is.

`range.strip`	a vector with two elements specifying the range of treatment effect size for display
`n.brk`	a number specifying the number of the points dividing the range of the argument "range.strip".
`n.brk.axis`	a number specifying the number of breakpoints dividing the axis of the argument "range.strip".
`font.size`	a vector specifying the size of labels and text; the first element is for the main title, the second is for for x-axis and y-axis labels; the thrid is for the legend text of subgroups; the fourth is for the subgroup labels near points; the fifth is for the unit labels on all the axes.
`title`	a string specifying the main title.
`lab.xy`	a list of two strings specifying the labels of the x and y axes.
`strip`	a string specifying the title of the colour strip.
`effect`	either "HR" or "RMST". only when outcome.type = "survival"
`equal.width`	A logical indicating whether the sectors should have equal width or proportional to their sample sizes
`show.KM`	A logical indicating whether to show the Kaplan-Meier curves for the subgroups
`show.effect`	A logical indicating whether to show the treatment effect
`conf.int`	A logical indicating whether to show confidence intervals for the treatment effect.
`show.overall`	A logical indicating whether to show the overall treatment effect and its confidence intervals in the reference strip
`palette`	either "divergent" or "hcl"
`col.power`	to be used when palette = "hcl". see colorspace package for reference

Examples

```
library(dplyr)

# Load the data to be used
data(prca)
dat <- prca
vars = data.frame(variable = names(dat), index = 1:length(names(dat)))
levels(dat$age_group) = c("Young","Middle-aged","Old")
levels(dat$weight_group)  = c("Low","Mid","High")
dat %>%
  rename(Age= age_group,
         Weight = weight_group)-> dat

set.seed(55643)
plot_circle(dat,
            covari.sel = c(14, 15),
            trt.sel = 3,
            resp.sel = c(1, 2),
            outcome.type = "survival",
            range.v = NULL, adj.ann.subgrp = 4,
```

```
            range.strip=c(-3, 3),
            n.brk = 31,
            n.brk.axis = 7,
            font.size = c(1, 1, 0.85, 0.85, 1),
            title = NULL, lab.xy = NULL,
            strip = "Treatment effect size (log hazard ratio)",
            effect = "HR",
            equal.width = FALSE,
            show.KM = FALSE,
            show.effect = TRUE,
            conf.int = FALSE, palette = "hcl")
```

`plot_circle2` *Circular plot using circlize package with matrix layout.*

Description

This function produces a circular plot for subgroup analysis using the circlize package. Overlaps across all variables are displayed in a matrix layout.

Usage

```
plot_circle2(
  dat,
  covari.sel,
  trt.sel,
  resp.sel,
  outcome.type,
  range.v = NULL,
  adj.ann.subgrp = 4,
  range.strip = c(-3, 3),
  n.brk = 31,
  n.brk.axis = NULL,
  font.size = c(1, 1, 0.85, 0.85, 1),
  title = NULL,
  lab.xy = NULL,
  strip = "Treatment effect size",
  effect = "HR",
  equal.width = TRUE,
  show.KM = FALSE,
  show.effect = TRUE,
  conf.int = TRUE,
  palette = "divergent",
  col.power = 0.5,
  nrow = 2,
  ncol = 4
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of the two covariates
`trt.sel`	a covariate index specifying the treatment code
`resp.sel`	a covariate index specifying the response variable
`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary" or "survival".
`range.v`	a vector specifying the vertical range of graphical display.
`adj.ann.subgrp`	a parameter adjusting the distance between a point and its corresponding subgroup label. The smaller the value is, the larger the distance is.
`range.strip`	a vector with two elements specifying the range of treatment effect size for display
`n.brk`	a number specifying the number of the points dividing the range of the argument "range.strip".
`n.brk.axis`	a number specifying the number of breakpoints dividing the axis of the argument "range.strip".
`font.size`	a vector specifying the size of labels and text; the first element is for the main title, the second is for for x-axis and y-axis labels; the thrid is for the legend text of subgroups; the fourth is for the subgroup labels near points; the fifth is for the unit labels on all the axes.
`title`	a string specifying the main title.
`lab.xy`	a list of two strings specifying the labels of the x and y axes.
`strip`	a string specifying the title of the colour strip.
`effect`	either "HR" or "RMST". only when outcome.type = "survival"
`equal.width`	A logical indicating whether the sectors should have equal width or proportional to their sample sizes
`show.KM`	A logical indicating whether to show the Kaplan-Meier curves for the subgroups
`show.effect`	A logical indicating whether to show the treatment effect
`conf.int`	A logical indicating whether to show confidence intervals for the treatment effect.
`palette`	either "divergent" or "hcl"
`col.power`	to be used when palette = "hcl". see colorspace package for reference
`nrow`	Number of rows in the matrix layout
`ncol`	Number of columns in the matrix layout

Examples

```
# Load the data to be used
data(prca)
dat <- prca
set.seed(12)
plot_circle2(dat,
```

```
                      covari.sel = c(4, 5, 6, 7),
                      trt.sel = 3,
                      resp.sel = c(1,2),
                      outcome.type = "survival",
                      range.v = NULL,
                      adj.ann.subgrp = 4,
                      range.strip=c(-3, 3),
                      n.brk = 31,
                      n.brk.axis = 7,
                      font.size = c(1, 1, 0.85, 0.85, 1),
                      title = NULL, lab.xy = NULL,
                      strip = "Treatment effect size (log hazard ratio)",
                      effect = "HR",
                      equal.width = FALSE,
                      show.KM = FALSE,
                      show.effect = TRUE,
                      conf.int = FALSE, palette = "hcl")
```

`plot_contour` *Contour plot for subgroup effect size*

Description

this function produces a contour plot showing the treatment effect size of subgroups. The subgroups are first defined by certain ranges of the first continuous covariate; and then further divided into smaller subgroup by certain ranges of the second covariate . The subgroups over the first covariate have a sample size close to one pre-specified value (N2) and any neighboring subgroups have an overlap size near the second pre-specified value (N1). Similarly, each subgroup over the first covariate has a sample size near the third pre-specified value (N4), and any neighboring subgroups which are further divided over the second covariate have a sample size near the fourth pre-specified value (N3). The x-coordinate and y-coordinate of a point indicates the middle point of the range over the first covariate and that over the second covariate, respectively. The contours show approximate effect sizes which are obtained by fitting grid points over the polynormial surface interpolating the points corresponding to subgroups.Note that there are three parameters for controlling the setting of contours. In addition, the function uses log odd ratio and log hazard ratio for displaying subgroup effect sizes in binary and survival data, respectively. Also, the actual subgroup sample sizes over the covariates are shown on the console window.

Usage

```
plot_contour(
  dat,
  covari.sel,
  trt.sel,
  resp.sel,
  outcome.type,
```

```
  setup.ss,
  n.grid = c(41, 41),
  brk.es = c(0, 1, 2, 3),
  n.brk.axis = 7,
  para.plot = c(0.35, 2, 20),
  font.size = c(1.5, 1.2, 1, 0.85, 0.8),
  title = NULL,
  subtitle = "default",
  effect = "HR",
  point.size = 1.2,
  filled = FALSE,
  strip = NULL,
  show.overall = FALSE,
  palette = "divergent",
  col.power = 0.5,
  show.points = FALSE,
  new.layout = TRUE
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of the two covariates
`trt.sel`	a variable index specifying the treatment code
`resp.sel`	a variable index specifying the response variable
`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary" or "survival".
`setup.ss`	a vector specifying approximate subgroup sample size and neibourghing subgroup overlap sample size. The first and the second elements are for overlap sizes and subgroup sample sizes over the first covariate; the third and thefourth are for further divided overlap sizes and subgroup sample sizes over the second covariate.
`n.grid`	a vector specifying the numbers of the grid points on the x-axis and the y-axis respectively.
`brk.es`	a vector specifying the break points on effect size, where each range partition is given with a different colour on points.
`n.brk.axis`	a number specifying the number of breakpoints dividing the axis of the argument "range.strip".
`para.plot`	a vector specifying the parameters of the contour plot; the first value is for controlling the degree of smoothing; the second is for controlling the degree of the polynomials fitting to be used (normally 1 or 2); the third is for controlling the number of contour lines.
`font.size`	a vector specifying the size of labels and text; the first element is for the main title, the second is for for x-axis and y-axis labels; the third is for the subtitle; the fourth is for the text in the legend; the fifth is for the labels on contour lines.
`title`	a string specifying the main title.

`subtitle`	strings specifying the subtitle
`effect`	either "HR" or "RMST". only when outcome.type = "survival"
`point.size`	size of points for raw data points
`filled`	a logical indicator whether to show filled contour plot. If FALSE, contour lines are drawn
`strip`	the title for the strip showing treatment effect size
`show.overall`	logical. whether to show or not the overall treatment effect in the strip
`palette`	either "divergent" or "hcl"
`col.power`	to be used when palette = "hcl". see colorspace package for reference
`show.points`	a logical indicator specifying whether to show the raw data points
`new.layout`	logical. If TRUE (default), the function calls graphics::layout(matrix(c(1, 2), nrow=1, ncol=2), widths=c(4,1)) to start from an empty page.

Examples

```
library(dplyr)

# Load the data to be used
data(prca)
dat <- prca
setup.ss =  c(10,60,15,30)
sub.title = bquote(N[11] %~~% .(setup.ss[2]) ~", "~
                     N[12] %~~% .(setup.ss[1]) ~", "~
                     N[21] %~~% .(setup.ss[4]) ~", "~
                     N[22] %~~% .(setup.ss[3]))
dat %>%
  rename(Weight = weight,
         Age = age) -> dat

plot_contour(dat,
             covari.sel = c(8,9),
             trt.sel = 3,
             resp.sel = c(1,2),
             outcome.type = "survival",
             setup.ss =  setup.ss,
             n.grid = c(100,100),
             brk.es = seq(-4.5,4.5,length.out = 101),
             n.brk.axis =  7,
             para.plot = c(0.5, 2, 6),
             font.size = c(1, 1, 1, 1, 1),
             title = NULL,
             subtitle = sub.title,
             strip = paste("Treatment effect size (log hazard ratio)"),
             show.overall = TRUE,show.points = TRUE,
             filled = TRUE, palette = "hcl",col.power = 0.75)
```

`plot_contour_localreg` *Contour plot for effect size via local regression.*

Description

this function produces a contour plot showing the treatment effect size of subgroups. The subgroups are first defined by certain ranges of the first continuous covariate; and then further divided into smaller subgroup by certain ranges of the second covariate . The subgroups over the first covariate have a sample size close to one pre-specified value (N2) and any neighboring subgroups have an overlap size near the second pre-specified value (N1). Similarly, each subgroup over the first covariate has a sample size near the third pre-specified value (N4), and any neighboring subgroups which are further divided over the second covariate have a sample size near the fourth pre-specified value (N3). The x-coordinate and y-coordinate of a point indicates the middle point of the range over the first covariate and that over the second covariate, respectively. The contours show approximate effect sizes which are obtained by fitting grid points over the polynormial surface interpolating the points corresponding to subgroups.Note that there are three parameters for controlling the setting of contours. In addition, the function uses log odd ratio and log hazard ratio for displaying subgroup effect sizes in binary and survival data, respectively. Also, the actual subgroup sample sizes over the covariates are shown on the console window.

Usage

```
plot_contour_localreg(
  dat,
  covari.sel,
  trt.sel,
  resp.sel,
  outcome.type,
  setup.ss,
  n.grid = c(41, 41),
  brk.es = c(0, 1, 2, 3),
  n.brk.axis = 3,
  para.plot = c(0.35, 2, 20),
  font.size = c(1.5, 1.2, 1, 0.85, 0.8),
  title = NULL,
  subtitle = NULL,
  unit.x = 1,
  unit.y = 1,
  effect = "HR",
  show.overall = TRUE,
  strip = "Effect Size",
  new.layout = TRUE
)
```

Arguments

`dat`	a data set

`covari.sel`	a vector of indices of the two covariates
`trt.sel`	a variable index specifying the treatment code
`resp.sel`	a variable index specifying the response variable
`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary" or "survival".
`setup.ss`	a vector specifying approximate subgroup sample size and neibourghing subgroup overlap sample size. The first and the second elements are for overlap sizes and subgroup sample sizes over the first covariate; the third and thefourth are for further divided overlap sizes and subgroup sample sizes over the second covariate.
`n.grid`	a vector specifying the numbers of the grid points on the x-axis and the y-axis respectively.
`brk.es`	a vector specifying the break points on effect size, where each range partition is given with a different colour on points.
`n.brk.axis`	a number specifying the number of breakpoints dividing the axis of the argument "range.strip".
`para.plot`	a vector specifying the parameters of the contour plot; the first value is for controlling the degree of smoothing; the second is for controlling the degree of the polynomials fitting to be used (normally 1 or 2); the third is for controlling the number of contour lines.
`font.size`	a vector specifying the size of labels and text; the first element is for the main title, the second is for for x-axis and y-axis labels; the third is for the subtitle; the fourth is for the text in the legend; the fifth is for the labels on contour lines.
`title`	a string specifying the main title.
`subtitle`	strings specifying the subtitle
`unit.x`	step for the x variable to create the grid that will center the kernel to apply the local regression
`unit.y`	step for the x variable to create the grid that will center the kernel to apply the local regression
`effect`	either "HR" or "RMST". only when outcome.type = "survival"
`show.overall`	logical. whether to show or not the overall treatment effect in the strip
`strip`	a string specifying the title of the colour strip.
`new.layout`	logical. If TRUE (default), the function calls graphics::layout(matrix(c(1, 2), nrow=1, ncol=2), widths=c(4,1)) to start from an empty page.

Examples

```
library(dplyr)

# Load the data to be used
data(prca)
dat <- prca
dat %>%
```

```
  rename(Weight = weight,
         Age = age) -> dat

plot_contour_localreg(dat,
                      covari.sel = c(8,9),
                      trt.sel = 3,
                      resp.sel = c(1,2),
                      n.grid = c(100,100),
                      font.size = c(1, 1, 1, 1, 1),
                      brk.es = seq(-4.5,4.5,length.out = 101),
                      n.brk.axis =  7,
                      strip = "Treatment effect size (log hazard ratio)",
                      outcome.type = "survival")
```

`plot_dissimilarity` *a plot for displaying dissimilarity distances of subgroups*

Description

this function produces a plot for displaying dissimilarity distances of pairwise subgroups, where dissmiliarity distance is defined by 1 - |intersect(A, B)|/|A|, for any sets A and B and A is the baseline set. The horizontal axis represents dissimilarity distance. The letters represent subgroups defined by the categories of the selected covariates. The letter above the green triangle is the baseline for calculating dissimilarity distances with the others above the red cross. There are two modes of display - one is that subgroup letters are located at the exact dissimilarity distance; the other is that letters are located at the middle of the category of dissimilarity distances. Note that some dissimilarity distances are known (such as 0 and 1) and therefore they are not shown in the graphical display. Also, the range of dissimilarity distances can be adjusted.

Usage

```
plot_dissimilarity(
  dat,
  covari.sel,
  mode,
  range.ds = c(0, 1),
  font.size = c(1, 0.9, 1, 0.7),
  title = NULL,
  lab.x = NULL
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of covariates
`mode`	a value specifying the type of display; either 1 or 2.

`range.ds`	a vector specifying the range of the dissimilarity distance
`font.size`	a vector specifying the size of labels and text; the first element is for the title; the second is for the x-axis label; the third is for the labels of baseline subgroups; the fourth is for the remaining subgroup labels (except for the baseline subgroup).
`title`	a string specifying the main titles.
`lab.x`	a string specifying the x-axis label.

Examples

```
data(prca)
dat <- prca

## 1. dissimilarity plot ---------------------------------------------------
plot_dissimilarity(dat = dat,
                   covari.sel = c(4,5,6),
                   mode = 3,
                   range.ds = c(0,1),
                   font.size = c(1, 0.9, 1, 0.7),
                   title = NULL,
                   lab.x = "Dissimilarity distance")
```

`plot_dissimilarity_alternative`

An alternative plot for displaying dissimilarity distances of subgroups

---

Description

this function produces a plot for displaying dissimilarity distances of pairwise subgroups, where dissmiliarity distance is defined by 1 - |intersect(A, B)|/|A|, for any sets A and B and A is the baseline set. The letters represent subgroups defined by the categories of the selected covariates.The vertical axis represents dissimilarity distance. Each column has a letter standing for a subgroup as the baseline for calculating dissimilarity distances. There are two modes of display - one is that subgroup letters are located at the exact dissimilarity distance with different types and colours of points; the other is that lines connected to all the points. The colour of lines or points indicates whether subgroups are from the same covariate or not. The number of types shows how many categories (from the same covariate) are considered as subgroups. The range of dissimilarity distances can be adjusted.

Usage

```
plot_dissimilarity_alternative(
  dat,
  covari.sel,
  mode,
  range.ds = c(0, 1),
  font.size = c(1.3, 1, 1),
```

```
  title = NULL,
  lab.y = NULL
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of covariates
`mode`	a value specifying the type of display; either 1 or 2.
`range.ds`	a vector specifying the range of the dissimilarity distance
`font.size`	a vector specifying the size of labels and text; the first element is for the title; the second is for the x-axis label; the third is for the labels of baseline subgroups; the fourth is for the remaining subgroup labels (except for the baseline subgroup).
`title`	a string specifying the main titles.
`lab.y`	a string specifying the y-axis label.

Examples

```
data(prca)
dat <- prca
## 1. dissimilarity alternative plot ---------------------------------------
plot_dissimilarity_alternative(dat = dat,
                               covari.sel = c(4,5,6),
                               mode = 2,
                               range.ds = c(0,1),
                               font.size = c(1, 1, 0.7),
                               title = NULL,
                               lab.y = "Similarity distance")
```

`plot_forest` *Forest plot for subgroup effect size*

Description

This function produces a forest plot showing the treatment effect size of subgroups defined by the categories of covariates. The first sub-figure provides a table of treatment effect estimate and sample size (for treatment / control group within each subgroup) ; the second sub-figure shows forest plots for subgroups and full population; the third displays forest plots of treatment and control group for each population. The dashed vertical line indicates no effect Note that the overall size of diamonds which represent subgroups can be adjusted by setting different values on the associated input argument. In addition, the function uses log odd ratio and log hazard ratio for displaying subgroup effect sizes in binary and survival data, respectively.

Usage

```
plot_forest(
  dat,
  covari.sel,
  trt.sel,
  resp.sel,
  outcome.type,
  size.shape = c(0.25, 0.12),
  font.size = c(1.3, 1, 0.85, 0.9),
  title = NULL,
  lab.x = NULL,
  time = mean(dat[, resp.sel[1]]),
  KM = FALSE,
  show.km.axis = TRUE,
  widths = c(1, 1, 1),
  max.time = NULL,
  n.brk = 10,
  trt.labels = c("Treatment", "Control"),
  panel.titles = c("Eff.size", "95% CI", "S.Size(T|C)"),
  eff.scale = c("logHR", "HR")
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of the two covariates
`trt.sel`	a covariate index specifying the treatment code
`resp.sel`	a covariate index specifying the response variable
`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary" or "survival".
`size.shape`	a vector specifying the height and width of the diamonds displaying sample sizes.
`font.size`	a vector specifying the size of labels and text; the first element is for the main titles, the second is for for the x-axis labels; the thrid is for the text in the first sub-figure; the fourth is for the unit labels of the x-axis.
`title`	a list of three strings specifying the main titles of the three sub-figures.
`lab.x`	a list of three strings specifying the x-axis labels of the three sub-figures.
`time`	time for calculating the RMST
`KM`	a logical indicating whether to show the Kaplan-Meier curves in the third panel
`show.km.axis`	a logical indicating whether to show the axes in the Kaplan-Meier curves
`widths`	a vector of length 3 indicating the widths of the panels
`max.time`	a numeric input indicating the maximum time for x-axis in the the Kaplan-Meier curves. If NULL, the maximum is taken from the dataset.
`n.brk`	number of breaks in the Kaplan-Meier curves

`trt.labels`	A character vector or length 2 specifying the labels of the treatments
`panel.titles`	A character vector or length 3 specifying the column titles in the first panel
`eff.scale`	Either "logHR" or "HR". Only necessary when outcome.type = "survival".

Examples

```
# Load the data to be used
library(dplyr)
data(prca)
dat <- prca
dat %>%
  mutate(bm = factor(ifelse(bm == 0 , "No", "Yes")),
         hx = factor(ifelse(hx == 0 , "No", "Yes"))) -> dat

## 5. Forest Plot -----------------------------------------------------------
main.title = list("", "Forest plot of subgroups",
                  "Kaplan-Meier curves\n by treatment group")
label.x = list("", "Log hazard ratio",
               "Time (days)")

plot_forest(dat,
            covari.sel = c(4,5,6,7),#vars
            trt.sel = 3,
            resp.sel = c(1, 2),
            outcome.type = "survival",
            size.shape = c(0.3, 6.5/4),
            font.size = c(0.6, 0.5, 0.4, 0.6),
            title = main.title,
            lab.x = label.x, time = 50, KM = TRUE,
            show.km.axis = 2, n.brk = 12, max.time = 77,
            widths = c(1,1,0.6))
```

`plot_labbe` *L'Abbe plot for subgroup effect size*

Description

this function produces a L'Abbe plot showing the treatment effect size of subgroups defined by the categories of covariates. The x-axis and y-axis represent the treatment effect estimate from the control and treatment group, respectively. The dashed dignal indicates no effect, and the solid diagnal line corresponds to the full population effect estimate. The squares represent subgroups, and the red or blue dashed lines have a length which show the magnitude of the lower or upper bound of the 95 treatment effect estimates. If the solid diagonal line cross all the blue or red lines, it may indicate homogeneity across subgroup with repective to the full population effect estimate. Note that the overall size of squares which represent subgroups can be adjusted by setting different values on the associated input argument. In addition, the function uses log odd ratio for displaying subgroup effect sizes in binary and survival data, respectively.

Usage

```
plot_labbe(
  dat,
  covari.sel,
  trt.sel,
  resp.sel,
  outcome.type,
  effect = c("HR", "RMST"),
  size.shape = 1/18,
  adj.ann.subgrp = 1/30,
  font.size = c(1, 1, 0.85, 0.85),
  title = NULL,
  lab.xy = NULL,
  time = mean(dat[, resp.sel[1]]),
  show.ci = TRUE,
  legend.position = c("inside", "outside")
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of the two covariates
`trt.sel`	a covariate index specifying the treatment code
`resp.sel`	a covariate index specifying the response variable
`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary".
`effect`	Either "HR" of "RMST". Only when outcome.type = "survival"
`size.shape`	a parameter controlling the height and width of the squares and the size is proportional to the ratio of subgroup sample sizes over the full population size.
`adj.ann.subgrp`	a parameter controlling the distance between a square and its corresponding subgroup label; also, the line gap between the legend text for subgroups.
`font.size`	a vector specifying the size of labels and text; the first element is for the main title, the second is for for x-axis and y-axis labels; the thrid is for the legend text of subgroups; the fourth is for the subgroup labels near the corresponding squares.
`title`	a string specifying the main title.
`lab.xy`	a list of two strings specifying the labels of the x and y axes.
`time`	time for calculating the survival in each subgroup
`show.ci`	A logical indicating whether to show an additional line for confidence intervals
`legend.position`	where to place the legend? either "inside" or "outside"

Examples

```
library(dplyr)

# Load the data to be used
data(prca)
dat <- prca
dat %>%
  mutate(bm = factor(ifelse(bm == 0 , "No", "Yes")),
         hx = factor(ifelse(hx == 0 , "No", "Yes")))-> dat

## 8. Labbe Plot -----------------------------------------------------------
lab.xy = list("Control Group Estimate", "Treatment Group Estimate")
plot_labbe(dat = dat,
           covari.sel = c(4,5,6,7),
           trt.sel = 3,
           resp.sel = c(1,2),
           outcome.type = "survival",
           effect = "RMST",
           lab.xy = lab.xy,
           size.shape = 0.2,
           adj.ann.subgrp = 1/30,
           time=50, show.ci = FALSE)
```

`plot_level` *Level plot for subgroup effect size*

Description

This function produces a level plot showing the treatment effect size of pairwise subgroups and marginal subgroups defined by the categories of two covariates. Also, it prints out the minimum and maximum of the treatment effect size on the console so as to set an approapriate range for effect size on the colour strip. Note that there are two types of graphical display; whether show subgroup sample size by rectangles with different sizes (proportional to the ratio of sample size to the full size) or not. In addition, the function uses log odd ratio and log hazard ratio for displaying subgroup effect sizes in binary and survival data, respectively.

Usage

```
plot_level(
  dat,
  covari.sel,
  trt.sel,
  resp.sel,
  outcome.type,
  ss.rect = FALSE,
  range.strip = c(-6, 6),
  n.brk = 30,
```

```
  n.brk.axis = NULL,
  font.size = c(15, 12, 0.8, 15, 0.6),
  title = NULL,
  strip = NULL,
  effect = c("HR", "RMST"),
  time = NULL,
  show.overall = FALSE,
  palette = "divergent",
  col.power = 0.5,
  grid.newpage = TRUE
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of the two covariates
`trt.sel`	a covariate index which specifies the treatment code
`resp.sel`	a covariate index which specifies the response variable
`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary" or "survival".
`ss.rect`	a logical operator displaying the rectangles for subgroup sample sizes if TRUE
`range.strip`	a vector with two elements specifying the range of treatment effect size for display
`n.brk`	a number specifying the number of the points dividing the range of the argument "range.strip".
`n.brk.axis`	a number specifying the number of breakpoints dividing the axis of the argument "range.strip".
`font.size`	a vector specifying the size of labels and text; the first element is for the main title; the second element is for the covariates labels and the colour strip label; the third is for the category labels of the first and second covariates; the fourth is for the text in the middle cells; the fifth is for the unit label on the colour strip.
`title`	a string specifying the main title.
`strip`	a string specifying the title of the colour strip.
`effect`	either "HR" or "RMST". only when outcome.type = "survival"
`time`	time for calculating the RMST
`show.overall`	logical. whether to show or not the overall treatment effect in the strip
`palette`	either "divergent" or "hcl"
`col.power`	to be used when palette = "hcl". see colorspace package for reference
`grid.newpage`	logical. If TRUE (default), the function calls grid::grid.newpage() to start from an empty page.

Examples

```
# Load the data to be used
data(prca)
dat  <- prca
levels(dat$age_group)    <- c("Young", "Middle-aged", "Old")
levels(dat$weight_group) <- c("Low", "Mid", "High")
names(dat)[c(14,15)]     <- c("age", "weight")

## 1.a Level plot ---------------------------------------------------------
plot_level(dat,
           covari.sel = c(14,15),
           trt.sel = 3,
           resp.sel = c(1, 2),
           outcome.type = "survival",
           ss.rect = FALSE,
           range.strip=c(-3, 3),
           n.brk = 31,
           n.brk.axis =  7,
           font.size = c(14, 12, .8, 14, 0.7),
           title = "Total sample size = 475",
           strip = "Treatment effect size (log hazard ratio)",
           effect = "HR",
           show.overall = TRUE, palette = "hcl")

## 1.b Modified Level plot ------------------------------------------------
plot_level(dat,
           covari.sel = c(14,15),
           trt.sel = 3,
           resp.sel = c(1, 2),
           outcome.type = "survival",
           ss.rect = TRUE,
           range.strip=c(-3, 3),
           n.brk = 31,
           n.brk.axis =  7,
           font.size = c(14, 12, .8, 14, 0.7),
           title = paste0("Total sample size = ", nrow(dat)),
           strip = "Treatment effect size (log hazard ratio)",
           show.overall = TRUE, palette = "hcl")
```

`plot_matrix_overlap` *matrix plots for relative overlap proportions of pairwise subgroups*

Description

this function produces a matrix plot for displaying relative proportions of pairwise subgroup overlap, where the relative proportion is defined by |intersect(A, B)|/|A|, for any sets A and B and A is the baseline set. The letters over (and beside) the matrix represent subgroups defined by the categories of the selected covariates. The subgroup letters on the rows are regarded as baseline subgroups for calculating relative overlap proportions. There are two modes of display - one with circles which are placed in the cells, where the size and the colour reflect how large the proportion is; the other with colours filled in the cells, where the colour indicate the level of overlap proportions.

Usage

```
plot_matrix_overlap(
  dat,
  covari.sel,
  mode,
  font.size = c(2, 1, 0.8),
  title = NULL
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of covariates
`mode`	a value specifying the type of display; either 1 or 2,
`font.size`	a vector specifying the size of labels and text; the first element is for the title; the second is for the covariates labels.
`title`	a string specifying the main title.

Examples

```
# Load the data to be used
data(prca)
dat <- prca

## 1. Matrix Overlap plot ----------------------------------------------------
plot_matrix_overlap(dat,
                    covari.sel = c(6,5,4,7),
                    mode = 1,
                    font.size = c(1.5, 1.25, 0.8),
                    title = NULL)
```

`plot_mosaic` *Mosaic plot*

Description

This function produces a mosaic plot for subgroup analysis

Usage

```
plot_mosaic(
  dat,
  covari.sel,
  trt.sel,
  resp.sel,
  outcome.type,
  range.v = NULL,
  adj.ann.subgrp = 4,
  range.strip = c(-3, 3),
  n.brk = 30,
  n.brk.axis = NULL,
  font.size = c(1, 1, 0.85, 0.85, 1),
  title = NULL,
  lab.xy = NULL,
  strip = "default",
  effect = "HR",
  lwd. = 2,
  sep. = 0.05,
  show.overall = TRUE,
  palette = "divergent",
  col.power = 0.5,
  print.ss = FALSE,
  col.line = "white",
  time = NULL,
  show.marginal = TRUE,
  show.effect = TRUE,
  grid.newpage = TRUE
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of the two covariates
`trt.sel`	a covariate index specifying the treatment code
`resp.sel`	a covariate index specifying the response variable
`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary" or "survival".
`range.v`	a vector specifying the vertical range of graphical display.
`adj.ann.subgrp`	a parameter adjusting the distance between a point and its corresponding subgroup label. The smaller the value is, the larger the distance is.
`range.strip`	range for the treatment effect scale
`n.brk`	number of breaks in the treatment effect scale
`n.brk.axis`	number of breaks in the axis of the treatment effect scale

`font.size`	a vector specifying the size of labels and text; the first element is for the main title, the second is for for x-axis and y-axis labels; the third is for the legend text of subgroups; the fourth is for the subgroup labels near points; the fifth is for the unit labels on all the axes.
`title`	a string specifying the main title.
`lab.xy`	a list of two strings specifying the labels of the x and y axes.
`strip`	title for the treatment effect scale
`effect`	either "HR" or "RMST". only when outcome.type = "survival"
`lwd.`	line width for the mosaics
`sep.`	separation for the mosaics
`show.overall`	logical. whether to show or not the overall treatment effect in the strip
`palette`	either "divergent" or "hcl"
`col.power`	to be used when palette = "hcl". see colorspace package for reference
`print.ss`	logival indicating whether to show the sample sizes of subgroups
`col.line`	color of the mosaics' border
`time`	time for calculating the RMST
`show.marginal`	logical indicating whether to show the marginal subgroups. only when 2 covariates are used
`show.effect`	logical indicating whether to show effect size using color or not. only when 3 covariates are used
`grid.newpage`	logical. If TRUE (default), the function calls grid::grid.newpage() to start from an empty page.

Examples

```
library(dplyr)
data(prca)
dat <- prca
dat %>%
  mutate(bm = factor(ifelse(bm == 0 , "No", "Yes")),
         hx = factor(ifelse(hx == 0 , "No", "Yes")),
         Treatment = factor(ifelse(rx == 0 , "Control", "Treatment")),
         Survival = factor(ifelse(survtime > 24 , "Yes", "No"),
         levels = c("Yes", "No")))-> dat
levels(dat$age_group) = c("Young","Middle-aged","Old")
levels(dat$weight_group)  = c("Low","Mid","High")
# Change variable names
dat %>%
  rename(`Bone Metastasis` = bm,
         `Performance rating` = pf,
         `History of cardiovascular events` = hx,
         `2-year survival` = Survival,
         Weight = weight_group,
         Age = age_group) -> dat
## 2.a Mosaic plot with 2 variables ----------------------------------------
plot_mosaic(dat = dat,
```

```
            covari.sel = c(14, 15),
            trt.sel = 3,
            resp.sel = c(1, 2),
            outcome.type =  "survival",
            range.v = NULL,
            adj.ann.subgrp = 4,
            range.strip=c(-3, 3),
            n.brk = 31,
            n.brk.axis = 7, sep. = 0.034,
            font.size = c(10, 10, 10, 10, 0.7),
            title = NULL, lab.xy = NULL,
            strip = "Treatment effect size (log-hazard ratio)",
            col.line = "white", lwd. = 2,
            effect = "HR", print.ss = FALSE, palette = "hcl")

## 2.b Mosaic plot with 3 variables ---------------------------------------
plot_mosaic(dat = dat,
            covari.sel = c(15, 7, 4),
            trt.sel = 3,
            resp.sel = c(1, 2),
            outcome.type =  "survival",
            range.v = NULL, adj.ann.subgrp = 4,
            range.strip=c(-3, 3),
            n.brk = 31, n.brk.axis = 7,
            font.size = c(10, 10, 10, 10, 0.7),
            title = NULL, lab.xy = NULL,
            strip = "Treatment effect size (log-hazard ratio)",
            effect = "HR", palette = "hcl")
```

`plot_network` *network overlap plot (only works for binary variables)*

Description

this function produces a plot for displaying relative proportions of pairwise subgroup overlap, where the relative proportion is defined by |intersect(A, B)|/|A|, for any sets A and B and A is the baseline set. The letters represent subgroups defined by the categories of the selected covariates. Subgroup letters located around a circle have two directional arrowed curved lines to the other. The arrowed lines have different colours, reflecting the level of overlap proportions. Note that if a subgroup is a baseline subgroup, then arrows on the curves would be located close to the corresponding letter. Also, the conditions of arrowed curves can be changed by setting different values on the argument "para".

Usage

```
plot_network(
  dat,
  covari.sel,
  para = c(0.2, 0.2, 1),
```

```
  font.size = c(1.5, 1.5, 0.8),
  title = NULL
)
```

Arguments

dat	a data set
covari.sel	a vector of indices of covariates
para	a vector with three elements specifying the parameters of plot display; the first element is for adjusting the curvature of curves; the second is for placing the relative position of arrowheads on the lines; the third is for specifying the adjustment of the arrowhead (there are three possible values, 0, 0.5 and 1)
font.size	a vector specifying the size of labels and text; the first element is for the title; the second is for the covariates labels.
title	a string specifying the main title.

Examples

```
# Load the data to be used
data(prca)
dat <- prca

## 1. Network plot -----------------------------------------------------------
plot_network(dat = dat,
             covari.sel = c(6,5,4,7),
             para = c(0.1, 0.5, 1),
             font.size = c(1.2, 1.2, 0.8),
             title = NULL)
```

`plot_nightingale` *Nightingale plot*

Description

This function produces a nightingale plot with a binary response.

Usage

```
plot_nightingale(
  dat,
  covari.sel,
  trt.sel = NULL,
  resp.sel,
  outcome.type = "binary",
  seq_by = 100,
  title = NULL,
```

```
  lab.y = NULL,
  strip = "Response",
  palette_colors = c("#faa8d2", "#80b1d3")
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of the two covariates
`trt.sel`	a covariate index specifying the treatment code
`resp.sel`	a covariate index specifying the response variable
`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary" or "survival".
`seq_by`	a number specifying the breaks in the y-axis
`title`	a string specifying the main title.
`lab.y`	a character value specifying the label of the y axis
`strip`	title for the legend
`palette_colors`	the colors for the graphics, a two character vector.

Examples

```
library(dplyr)
dat <- prca
levels(dat$age_group) = c("Young","Middle-aged","Old")
levels(dat$weight_group)  = c("Low","Mid","High")
comb_levels = c("Young - Low", "Young - Mid", "Young - High",
                "Middle-aged - Low", "Middle-aged - Mid", "Middle-aged - High",
                "Old - Low", "Old - Mid", "Old - High")
dat %>%
  mutate(AgeWeight = factor(sprintf("%s - %s", age_group, weight_group),
                            levels = comb_levels))  %>%
  mutate(survival = factor(ifelse(survtime > 24 , "Yes", "No"),
                           levels = c("No", "Yes"))) %>%
  mutate(rx = factor(rx, labels = c("Control", "Treatment"))) -> dat

plot_nightingale(dat = dat, covari.sel = 16, resp.sel = 17,
                 strip = "2-year survival")
plot_nightingale(dat = dat, trt.sel = 3, covari.sel = 16,
                 resp.sel = 17,
                 seq_by = 50,
                 strip = "2-year survival")
```

`plot_nightingale_effect`

Nightingale plot with treatment effects

Description

This function produces a nightingale plot with treatment effects across subgroups

Usage

```
plot_nightingale_effect(
  dat,
  covari.sel,
  trt.sel,
  resp.sel,
  outcome.type,
  seq_by = 100,
  range.strip = c(-6, 6),
  n.brk = 30,
  n.brk.axis = NULL,
  title = NULL,
  strip = NULL,
  effect = c("HR", "RMST"),
  time = NULL,
  show.overall = FALSE,
  lab.y = NULL,
  palette = "hcl",
  col.power = 0.5
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of the two covariates
`trt.sel`	a covariate index specifying the treatment code
`resp.sel`	a covariate index specifying the response variable
`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary" or "survival".
`seq_by`	a number specifying the breaks in the y-axis
`range.strip`	a vector with two elements specifying the range of treatment effect size for display
`n.brk`	a number specifying the number of the points dividing the range of the argument "range.strip".
`n.brk.axis`	a number specifying the number of breakpoints dividing the axis of the argument "range.strip".

`title`	a string specifying the main title.
`strip`	a string specifying the title of the colour strip.
`effect`	either "HR" or "RMST". only when outcome.type = "survival"
`time`	time for calculating the RMST
`show.overall`	A logical indicating whether to show the overall treatment effect and its confidence intervals in the reference strip
`lab.y`	a character value specifying the label of the y axis
`palette`	either "divergent" or "hcl"
`col.power`	to be used when palette = "hcl". see colorspace package for reference

Examples

```
data(prca)
dat <- prca
vars <- data.frame(variable = names(dat), index = 1:length(names(dat)))
levels(dat$age_group) <- c("Young", "Middle-aged", "Old")
levels(dat$weight_group) <- c("Low", "Mid", "High")
names(dat)[c(14,15)] <- c("Age", "Weight")

strip.title = "Treatment effect size (log hazard ratio)"
plot_nightingale_effect(dat,
                        covari.sel = c(14,15),
                        trt.sel = 3,
                        resp.sel = c(1, 2),
                        outcome.type = "survival",
                        range.strip=c(-3, 3),
                        n.brk = 31,
                        n.brk.axis =  7,
                        title = "Total sample size = 475",
                        strip = strip.title, effect = "HR",
                        show.overall = TRUE, palette = "hcl")
```

`plot_overlap`

A graphical display for showing relative proportions of pairwise subgroup overlap with two unidirectional arrowed curved lines around a circle

Description

This function produces a plot for displaying relative proportions of pairwise subgroup overlap, where the relative proportion is defined by |intersect(A, B)|/|A|, for any sets A and B and A is the baseline set. The letters represent subgroups defined by the categories of the selected covariates. Subgroup letters located around a circle have two directional arrowed curved lines to the other. The arrowed lines have different colours, reflecting the level of overlap proportions. Note that if a subgroup is a baseline subgroup, then arrows on the curves would be located close to the corresponding letter. Also, the conditions of arrowed curves can be changed by setting different values on the argument "para". Contrary to `plot_overlap2`, this function produces a single diagram.

Usage

```
plot_overlap(
  dat,
  covari.sel,
  para = c(0.2, 0.2, 1),
  font.size = c(1.5, 1.5, 1),
  title = NULL
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of covariates
`para`	a vector with three elements specifying the parameters of plot display; the first element is for adjusting the curvature of curves; the second is for placing the relative position of arrowheads on the lines; the third is for specifying the adjustment of the arrowhead (there are three possible values, 0, 0.5 and 1)
`font.size`	a vector specifying the size of labels and text; the first element is for the title; the second is for the covariates labels.
`title`	a string specifying the main title.

Examples

```
# Load the data to be used
data(prca)
dat <- prca

## Overlap plot --------------------------------------------------------------
plot_overlap(dat = dat,
             covari.sel = c(6,5,4,7),
             para = c(0.1, 0.5, 1),
             font.size = c(1.2, 1.2, 0.8),
             title = NULL)
```

`plot_overlap2` *a graphical display for showing relative proportions of pairwise subgroup overlap in a matrix layout*

Description

this function produces a plot for displaying relative proportions of pairwise subgroup overlap, where the relative proportion is defined by |intersect(A, B)|/|A|, for any sets A and B and A is the baseline set. The letters represent subgroups defined by the categories of the selected covariates. Subgroup are located around a circle have a directional arrowed curved line to the other subgroups. The arrowed lines have different colours, reflecting the level of overlap proportions. Note that if a subgroup is a baseline subgroup, then arrows on the curves would be located close to the corresponding letter. Also, the conditions of arrowed curves can be changed by setting different values on the argument "para". Contrary to `plot_overlap`, this function produces a matrix layout.

Usage

```
plot_overlap2(
  dat,
  covari.sel,
  para = c(0.2, 0.2, 1),
  font.size = c(1.5, 1.5, 0.8),
  title = NULL
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of covariates
`para`	a vector with three elements specifying the parameters of plot display; the first element is for adjusting the curvature of curves; the second is for placing the relative position of arrowheads on the lines; the third is for specifying the adjustment of the arrowhead (there are three possible values, 0, 0.5 and 1)
`font.size`	a vector specifying the size of labels and text; the first element is for the title; the second is for the covariates labels.
`title`	a string specifying the main title.

Examples

```
# Load the data to be used
data(prca)
dat <- prca

plot_overlap2(dat = dat,
             covari.sel = c(6,5,4,7),
             para = c(0.05, 0.75, 1),
             font.size = c(1.2, 1.2, 0.8),
             title = NULL)
```

`plot_overlap_alternative` *a graphical display for showing relative proportions of pairwise subgroup overlap with two unidirectional arrowed lines around two circle*

Description

this function produces a plot for displaying relative proportions of pairwise subgroup overlap, where the relative proportion is defined by |intersect(A, B)|/|A|, for any sets A and B and A is the baseline set. The letters represent subgroups defined by the categories of the selected covariates. Subgroup letters located around the two circles are considered as baselines; each has unidirectional arrowed lines to the other. In addition, there are two modes of display - one is that arrowed lines have different colours, reflecting the level of overlap proportions; the other is that lines have five thickness and types, indicating five levels of overlap proportions.

Usage

```
plot_overlap_alternative(
  dat,
  covari.sel,
  para = c(0.5, 0.15, 1),
  mode,
  font.size = c(1.5, 1.5, 0.8),
  title = NULL
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of covariates
`para`	a vector with three elements specifying the parameters of plot display; the first element is for specifying the length of the arrowhead; the second is for specifying the width of the arrowheads; the third is for specifying the adjustment of the arrowhead (there are three possible values, 0, 0.5 and 1)
`mode`	a value specifying the type of display; either 1 or 2.
`font.size`	a vector specifying the size of labels and text; the first element is for the title; the second is for the covariates labels.
`title`	a string specifying the main title.

Examples

```
data(prca)
dat <- prca
## 1. Overlap alternative plot ------------------------------------------------
plot_overlap_alternative(dat = dat,
                         covari.sel = c(6,5,4,7),
                         mode = 1,
                         para = c(0, 0.6, 1),
                         font.size = c(1.2, 1.2, 0.8),
                         title = NULL)
```

`plot_radial` *Radial (Galbraith) plot for subgroup effect size*

Description

This function produces a modified Galbraith's radial plot showing the treatment effect size of subgroups defined by the categories of covariates. The x-axis represents the reciprocal of the standard error of subgroup treatment effect estimates.The y-axis means standardized effect size difference (the difference between subgroup effect the full popultion effect is divided by the standard error of the estimator for the overall population effect. Points here are for subgroups. The grey region indicates whether subgroup effects are homogeneous to the full population effect or not. The two arcs on the right side show subgroup treatment effects in the original scale, where the red spots are the projection of points from the origin on the left side. Note that the vertical range of display can be changed by setting different values on the associated input argument. In addition, the function uses log odd ratio and log hazard ratio for displaying subgroup effect sizes in binary and survival data, respectively.

Usage

```
plot_radial(
  dat,
  covari.sel,
  trt.sel,
  resp.sel,
  outcome.type,
  range.v = NULL,
  adj.ann.subgrp = 4,
  font.size = c(1, 1, 0.85, 0.85, 1),
  title = NULL,
  lab.xy = NULL,
  legend.position = c("inside", "outside")
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of the two covariates
`trt.sel`	a covariate index specifying the treatment code
`resp.sel`	a covariate index specifying the response variable
`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary" or "survival".
`range.v`	a vector specifying the vertical range of graphical display.
`adj.ann.subgrp`	a parameter adjusting the distance between a point and its corresponding subgroup label. The smaller the value is, the larger the distance is.
`font.size`	a vector specifying the size of labels and text; the first element is for the main title, the second is for for x-axis and y-axis labels; the thrid is for the legend text of subgroups; the fourth is for the subgroup labels near points; the fifth is for the unit labels on all the axes.
`title`	a string specifying the main title.
`lab.xy`	a list of two strings specifying the labels of the x and y axes.
`legend.position`	where to place the legend? either "inside" or "outside"

Examples

```
library(dplyr)

# Load the data to be used
data(prca)
dat <- prca
dat %>%
  mutate(bm = factor(ifelse(bm == 0 , "No", "Yes")),
         hx = factor(ifelse(hx == 0 , "No", "Yes")))-> dat

label.xy = list(expression(1/hat(sigma)[hat(delta)[i]]),
                expression((hat(delta)[i]-hat(delta)[F])/hat(sigma)[hat(delta)[i]]))
plot_radial(dat,
            covari.sel = c(4, 5, 6, 7),
            trt.sel = 3,
            resp.sel = c(1, 2),
            outcome.type = "survival",
            range.v = c(-7, 6),
            adj.ann.subgrp = 4,
            font.size = c(1, 1, 0.75, 0.8, 0.85),
            lab.xy = label.xy)
```

`plot_radial2` *Modified Radial (Galbraith) plot for subgroup effect size*

Description

This function produces a modified Galbraith's radial plot showing the treatment effect size of subgroups defined by the categories of covariates. The x-axis represents the reciprocal of the standard error of the difference between the subgroup treatment effect estimates and the overall treatment effect estimate. The y-axis is the standardized effect size difference (the difference between subgroup effect the full popultion effect is divided by its standard error). Points here are for subgroups. The grey region indicates whether subgroup effects are homogeneous to the full population effect or not. The two arcs on the right side show subgroup treatment effects in the original scale, where the red spots are the projection of points from the origin on the left side. Note that the vertical range of display can be changed by setting different values on the associated input argument. In addition, the function uses log odd ratio and log hazard ratio for displaying subgroup effect sizes in binary and survival data, respectively.

Usage

```
plot_radial2(
  dat,
  covari.sel,
  trt.sel,
  resp.sel,
  outcome.type,
  range.v = NULL,
```

```
  adj.ann.subgrp = 4,
  font.size = c(1, 1, 0.85, 0.85, 1),
  title = NULL,
  lab.xy = NULL,
  plot.full = FALSE,
  legend.position = c("inside", "outside")
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of the two covariates
`trt.sel`	a covariate index specifying the treatment code
`resp.sel`	a covariate index specifying the response variable
`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary" or "survival".
`range.v`	a vector specifying the vertical range of graphical display.
`adj.ann.subgrp`	a parameter adjusting the distance between a point and its corresponding subgroup label. The smaller the value is, the larger the distance is.
`font.size`	a vector specifying the size of labels and text; the first element is for the main title, the second is for for x-axis and y-axis labels; the thrid is for the legend text of subgroups; the fourth is for the subgroup labels near points; the fifth is for the unit labels on all the axes.
`title`	a string specifying the main title.
`lab.xy`	a list of two strings specifying the labels of the x and y axes.
`plot.full`	a logical indicating whether to show the overall treatment effect
`legend.position`	where to place the legend? either "inside" or "outside"

Examples

```
library(dplyr)

# Load the data to be used
data(prca)
dat <- prca
dat %>%
  mutate(bm = factor(ifelse(bm == 0 , "No", "Yes")),
         hx = factor(ifelse(hx == 0 , "No", "Yes")))-> dat

lab.x = expression(1/sqrt(hat(Var)(hat(delta)[i]-hat(delta)[F])))
lab.y = expression((hat(delta)[i]-hat(delta)[F])/sqrt(hat(Var)(hat(delta)[i]-hat(delta)[F])))
lab.xy = label.xy = list(lab.x, lab.y)
plot_radial2(dat,
             covari.sel = c(4, 5, 6, 7),
             trt.sel = 3,
             resp.sel = c(1, 2),
```

```
                outcome.type = "survival",
                range.v = c(-11, 8),
                adj.ann.subgrp = 4,
                font.size = c(1, 1, .6, .6, .6),
                lab.xy = label.xy)
```

`plot_stepp` *STEPP for subgroup effect size*

Description

this function produces a plot of using the approach "Subpopulation Treatment Effect Pattern Plot". It shows the treatment effect size of subgroups, where subgruops are defined by certain ranges of a continuous covariate; each subgroup has a sample size close to a pre-specified value (N2) and any neighboring subgroups have an overlap size near another pre-specified value (N1). The plot shows the 95 a horizontal line). The y-coordinate of a point indicates the effect size within the corresponding subgroup; the x-coordinate shows the lower bound of the range which defines the subgroup. If part of the horizontal line is out of the simultaneous C.I., it may reveal hetergeneity across subgroup effects with repective to the overall effect. In addition, one can control the width of the C.I. by controlling Type I error rate in one of the function arguments. Note that the function uses log odd ratio and log hazard ratio for displaying subgroup effect sizes in binary and survival data, respectively. The actual subgroup sample sizes over the covariate are shown on the console window as well.

Usage

```
plot_stepp(
  dat,
  covari.sel,
  trt.sel,
  resp.sel,
  outcome.type,
  setup.ss,
  alpha,
  font.size = c(1.2, 1, 1, 0.85),
  title = NULL,
  lab.y = NULL,
  subtitle = NULL
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of the two covariates
`trt.sel`	a variate index specifying the treatment code
`resp.sel`	a variate index specifying the response variable

`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary" or "survival".
`setup.ss`	a vector specifying the approximate overlap size (N2) and subgroup sample size (N1).
`alpha`	the type I error rate
`font.size`	a vector specifying the size of labels and text; the first element is for main titles, the second is for for x-axis and y-axis labels; the thrid is for the text in the legend; the fourth is for the subtitle.
`title`	a string specifying the main title.
`lab.y`	a string specifying the labels of the y-axis.
`subtitle`	strings specifying the subtitle

Details

Contrary to `ggplot_stepp`, `plot_stepp` implements an x-axis where each midpoint of the subgroups is drawn equidistant.

See Also

`ggplot_stepp`

Examples

```
# Load the data to be used
data(prca)
dat <- prca

## 9. stepp Plot -----------------------------------------------------------
lab.y.title = paste("Treatment effect size (log-hazard ratio)");
setup.ss = c(30,40)
sub.title = paste0("(Subgroup sample sizes are set to ", setup.ss[2],
                   "; overlap sizes are set to ", setup.ss[1], ")" )
plot_stepp(dat,
           covari.sel = 8,
           trt.sel = 3,
           resp.sel = c(1, 2),
           outcome.type = "survival",
           setup.ss = c(30,40),
           alpha = 0.05,
           font.size = c(0.9, 1, 1, 1),
           title = NULL,
           lab.y = lab.y.title,
           subtitle = sub.title)
```

`plot_tree` *Tree plot for subgroup effect size*

Description

This function produces a tree plot showing the treatment effect size of subgroups defined by the categories of covariates. The left side shows treatment effect size; the right side indicates what covariate is considered. Each level shows the 95 effect estimate and subgroup sample sizes (by the width of horizontal violet lines). Each subgroup is further divided into several subgroups by categories of the covariate on the lower level. The horizontal line corresponding to the overal effect can be added into each level so as to check homogeneity across subgroup effects with repective to the overall effect. In addition, the function uses log odd ratio and log hazard ratio for displaying subgroup effect sizes in binary and survival data, respectively.

Usage

```
plot_tree(
  dat,
  covari.sel,
  trt.sel,
  resp.sel,
  outcome.type,
  add.aux.line = FALSE,
  font.size = c(15, 10, 0.5),
  title = NULL,
  lab.y = NULL,
  text.shift = 0.005,
  keep.y.axis = FALSE,
  grid.newpage = TRUE
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of the two covariates
`trt.sel`	a covariate index specifying the treatment code
`resp.sel`	a covariate index specifying the response variable
`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary" or "survival".
`add.aux.line`	a logical operator displaying the auxiliary horizontal line for checking heterogeneity in treatment effects if TRUE
`font.size`	a vector specifying the size of labels and text; the first element is for the main title and the second element is for the text in the left, right and bottom labels; the third is for the unit labels on the y-axis.
`title`	a string specifying the main title.

`lab.y`	a string specifying the y-axis label
`text.shift`	a numeric indicating the separation of the text in the branches
`keep.y.axis`	a logical indicating whether to keep the y axis fixed across the levels
`grid.newpage`	logical. If TRUE (default), the function calls grid::grid.newpage() to start from an empty page.

Examples

```
library(dplyr)

# Load the data to be used
data(prca)
dat <- prca
dat %>%
  mutate(bm = factor(ifelse(bm == 0 , "No", "Yes")),
         hx = factor(ifelse(hx == 0 , "No", "Yes"))) -> dat

## Tree plot with fixed y-axis
plot_tree(dat,
          covari.sel = c(4, 5, 7),
          trt.sel = 3,
          resp.sel = c(1, 2),
          outcome.type = "survival",
          add.aux.line = TRUE,
          font.size = c(12, 8, 0.55),
          title = NULL,
          lab.y = "Effect size (log hazard ratio)",
          keep.y.axis = TRUE)

## Tree plot with free y-axes
plot_tree(dat,
          covari.sel = c(4, 5, 7),
          trt.sel = 3,
          resp.sel = c(1, 2),
          outcome.type = "survival",
          add.aux.line = TRUE,
          font.size = c(12, 8, 0.55),
          title = NULL,
          lab.y = "Effect size (log hazard ratio)",
          keep.y.axis = FALSE)
```

`plot_venn` *Venn diagram for subgroup effect size*

Description

This function produces a Venn diagram showing the treatment effect size of subgroups defined by sets from the categories of covariates. Also, it prints out the minimum and maximum of the treatment effect size on the console so as to set an approapriate range for effect size on the colour strip . Note that there are two options of graphical display; whether show the subgroup effect size of the complement of the union of all the considered subgroups or not. In addition, this function only works up to 5 sets and does not run an area-proportional algorithms for displaying two or three set. In addition, the function uses log odd ratio and log hazard ratio for displaying subgroup effect sizes in binary and survival data, respectively.

Usage

```
plot_venn(
  dat,
  covari.sel,
  cat.sel,
  trt.sel,
  resp.sel,
  outcome.type,
  fill = TRUE,
  range.strip = c(-6, 6),
  n.brk = 13,
  n.brk.axis = 7,
  effect = "HR",
  show.overall = TRUE,
  palette = "divergent",
  col.power = 0.5,
  font.size = c(1, 1.5, 1, 0.9, 1, 1),
  title = NULL,
  strip = NULL,
  prop_area = FALSE,
  cat.dist = "default",
  grid.newpage = TRUE
)
```

Arguments

`dat`	a data set
`covari.sel`	a vector of indices of covariates
`cat.sel`	a vector of indices of the categories for each covariate
`trt.sel`	a covariate index specifying the treatment code
`resp.sel`	a covariate index specifying the response variable
`outcome.type`	a string specifying the type of the response variable, it can be "continuous", or "binary" or "survival".
`fill`	A logical indicating whether to use color for treatment effects
`range.strip`	a vector with two elements specifying the range of treatment effect size for display
`n.brk`	a number specifying the number of the points dividing the range of the argument "range.strip".

`n.brk.axis`	a number specifying the number of breakpoints dividing the axis of the argument "range.strip".
`effect`	either "HR" or "RMST". only when outcome.type = "survival"
`show.overall`	logical. whether to show or not the overall treatment effect in the strip
`palette`	either "divergent" or "hcl"
`col.power`	to be used when palette = "hcl". see colorspace package for reference
`font.size`	a vector specifying the size of labels and text; the first element is for the main title; the second is for the category labels; the third is for the sample size labels; the fourth is for the legend text; the fifth is for the y-axis label of the colour strip; the sixth is for the unit label on the y axis.
`title`	a string specifying the main title.
`strip`	a string specifying the title of the colour strip.
`prop_area`	A logical indicating whether to make the areas approximately proportional to the set size
`cat.dist`	A vector (length same as covari.sel) giving the distances (in npc units) of the category names from the edges of the circles (can be negative)
`grid.newpage`	logical. If TRUE (default), the function calls grid::grid.newpage() to start from an empty page.

Examples

```
library(dplyr)
# Load the data to be used
data(prca)

## 3.a Venn Diagram ------------------------------------------------------------
dat <- prca
dat %>%
  rename(Performance = pf,
         `Bone\nmetastasis` = bm,
         `History of\ncardiovascular\nevents` = hx) -> dat
plot_venn(dat,
          covari.sel = c(5, 7, 4),
          cat.sel = c(2,2,2),
          trt.sel = 3,
          resp.sel = c(1,2),
          outcome.type = "survival",
          fill = FALSE,
          cat.dist = c(0.04,0.04,0.07),
          font.size = c(0.5, 0.5, 0.7, 0.5, 0.6, 0.6))

## 3.b Filled Venn Diagram -----------------------------------------------------
dat <- prca
dat$age1 = factor(dat$age1)
dat %>%
  rename(Stage = stage,
         Performance = pf,
         `Bone\nmetastasis` = bm,
```

```
         `History of\ncardiovascular\nevents` = hx) -> dat
plot_venn(dat,
          covari.sel = c(4,6,7,5),#vars,
          cat.sel = c(2,2,2,2),
          trt.sel = 3,
          resp.sel = c(1,2),
          outcome.type = "survival",
          fill = TRUE,
          range.strip = c(-3, 3),
          n.brk = 31, n.brk.axis = 7,
          font.size = c(0.5, 0.5, 0.7, 0.5, 0.6, 0.6),
          strip = paste("Treatment effect size (log hazard ratio)"),
          palette = "hcl")


## 3.c Area-proportional Venn Diagram -------------------------------------------------------------
dat <- prca
plot_venn(dat,
          covari.sel = c(5,7,4),
          cat.sel = c(2,2,2),
          trt.sel = 3,
          resp.sel = c(1,2),
          outcome.type = "survival",
          fill = TRUE,
          range.strip = c(-3, 3),
          n.brk = 31, n.brk.axis = 7,
          font.size = c(0.5, 0.5, 0.7, 0.5, 0.6, 0.6),
          strip = paste("Treatment effect size (log hazard ratio)"),
          palette = "hcl", prop_area = TRUE)
```

`prca` *Prostate Cancer Dataset*

Description

a prostate carcinoma dataset from a clinical trial David P. Byar and Sylvan B. Green. The choice of treatment for cancer patients based on covariate information: application to prostate cancer. Bulletin du Cancer, 67:477–490, 1980. The data can be found in the web: `http://portal.uni-freiburg.de/imbi/Royston-Sauerbrei-book/index.html`. We downloaded it form the supplementary material of Rosenkranz (2016) https://onlinelibrary.wiley.com/doi/full/10.1002/bimj.201500147 modified the data to keep relevant variables, and created categorical ones from age and weight.

Usage

```
data(prca)
```

Format

A data frame with 475 rows and 15 variables:

survtime survival time. Response variable

cens The status indicator, 0=alive, 1=dead

rx treatment received, 0=control, 1=treatment

bm existence of bone metastasis

hx history of cardiovascular events

stage disease stage (3 or 4)

pf performance

age age

weight weight in kg.

age1 dummy variable for 65 <= age < 75

age2 dummy variable for age >= 75

wt1 dummy variable for 90 <= weight < 110

wt2 dummy variable for weight >= 110

age_group age categorized in 3 groups

weight_group weight categorized in 3 groups

Source

`http://portal.uni-freiburg.de/imbi/Royston-Sauerbrei-book/index.html`

`https://onlinelibrary.wiley.com/doi/full/10.1002/bimj.201500147`

Examples

```
## Not run:
# Code used to download the dataset and create variables
library(haven)
l1 <- "https://onlinelibrary.wiley.com/action/"
l2 <- "downloadSupplement?doi=10.1002%2Fbimj.201500147&attachmentId=2173117089"
data_url <- paste0(l1,l2)
temp <- tempfile()
download.file(data_url,temp)
prca0 <- read_sas(unz(temp, "adv_prostate_ca.sas7bdat"))
# Select the variables that we use for the analysis
prca <- prca0[,c("SURVTIME","CENS","RX","BM","HX","STAGE","PF", "AGE", "WT")]

# Change names of variables to lower case
names(prca)<- c("survtime","cens","rx","bm",
                "hx","stage","pf","age", "wt")

# Create subgroups for Age and Weight and Stage
prca$age1 <- 1 * (prca$age > 65 & prca$age <= 75)
prca$age2 <- 1 * (prca$age > 75)
prca$wt1  <- 1 * (prca$wt > 90 & prca$wt <= 110)
```

```
prca$wt2  <- 1 * (prca$wt > 110)

# Create subgroups for Age and Weight and Stage with (-1,1) coding
prca$agegroup <- 1 + (1 * (prca$age > 65) + 1 * (prca$age > 75))
prca$wtgroup  <- 1 + (1 * (prca$wt > 90) + 1 * (prca$wt > 110))
dat = prca
dat$agegroup = factor(dat$agegroup)
dat$wtgroup = factor(dat$wtgroup)
range(dat$age)
range(dat$wt)
levels(dat$agegroup) = c("[48,65]","(65,75]","(75,89]")
levels(dat$wtgroup)  = c("[69,90]","(90,110]","(110,152]")
## We need variables as factors
dat$bm    = factor(dat$bm)
dat$hx    = factor(dat$hx)
dat$stage = factor(dat$stage)
dat$pf    = factor(dat$pf)
dat$rx    = factor(dat$rx) # Treatment

# Put labels to the variables so that they appear in the plot
names(dat)<- c("survtime",
               "cens",
               "rx",
               "bm",
               "hx",
               "stage",
               "pf",
               "age",
               "weight",
               "age1",
               "age2",
               "wt1",
               "wt2",
               "age_group",
               "weight_group")
prca <- dat
## devtools::use_data(prca, overwrite = T) ## Use it as dataset for the package

## End(Not run)
```

`subgroupset` *UpSetR Plot with treatment effects*

Description

Visualization of set intersections using novel UpSet matrix design, modified to show treatment effects and confidence intervals

Usage

```
subgroupset(
  data,
  nsets = 5,
  nintersects = 40,
  sets = NULL,
  keep.order = F,
  set.metadata = NULL,
  intersections = NULL,
  matrix.color = "gray23",
  main.bar.color = "gray23",
  mainbar.y.label = "Intersection Size",
  mainbar.y.max = NULL,
  sets.bar.color = "gray23",
  sets.x.label = "Set Size",
  point.size = 2.2,
  line.size = 0.7,
  mb.ratio = c(0.7, 0.3),
  expression = NULL,
  att.pos = NULL,
  att.color = main.bar.color,
  order.by = c("freq", "degree"),
  decreasing = c(T, F),
  show.numbers = "yes",
  number.angles = 0,
  group.by = "degree",
  cutoff = NULL,
  queries = NULL,
  query.legend = "none",
  shade.color = "gray88",
  shade.alpha = 0.25,
  matrix.dot.alpha = 0.5,
  empty.intersections = NULL,
  color.pal = 1,
  boxplot.summary = NULL,
  effects.summary = NULL,
  outcome.type = c("continuous", "binary", "survival"),
  attribute.plots = NULL,
  scale.intersections = "identity",
  scale.sets = "identity",
  text.scale = 1,
  set_size.angles = 0,
  treatment.var = NULL,
  min.n = 20,
  icon = c("dots", "pm", "pm.circle", "value"),
  fill.trt = TRUE,
  transpose = FALSE
)
```

Arguments

`data`	Data set
`nsets`	Number of sets to look at
`nintersects`	Number of intersections to plot. If set to NA, all intersections will be plotted.
`sets`	Specific sets to look at (Include as combinations. Ex: c("Name1", "Name2"))
`keep.order`	Keep sets in the order entered using the sets parameter. The default is FALSE, which orders the sets by their sizes.
`set.metadata`	Metadata that offers insight to an attribute of the sets. Input should be a data frame where the first column is set names, and the remaining columns are attributes of those sets. To learn how to use this parameter it is highly suggested to view the set metadata vignette. The link can be found on the package's GitHub page.
`intersections`	Specific intersections to include in plot entered as a list of lists. Ex: list(list("Set name1", "Set name2"), list("Set name1", "Set name3")). If data is entered into this parameter the only data shown on the UpSet plot will be the specific intersections listed.
`matrix.color`	Color of the intersection points
`main.bar.color`	Color of the main bar plot
`mainbar.y.label`	The y-axis label of the intersection size bar plot
`mainbar.y.max`	The maximum y value of the intersection size bar plot scale. May be useful when aligning multiple UpSet plots horizontally.
`sets.bar.color`	Color of set size bar plot
`sets.x.label`	The x-axis label of the set size bar plot
`point.size`	Size of points in matrix plot
`line.size`	Width of lines in matrix plot
`mb.ratio`	Ratio between matrix plot and main bar plot (Keep in terms of hundreths)
`expression`	Expression to subset attributes of intersection or element query data. Enter as string (Ex: "ColName > 3")
`att.pos`	Position of attribute plot. If NULL or "bottom" the plot will be at below UpSet plot. If "top" it will be above UpSert plot
`att.color`	Color of attribute histogram bins or scatterplot points for unqueried data represented by main bars. Default set to color of main bars.
`order.by`	How the intersections in the matrix should be ordered by. Options include frequency (entered as "freq"), degree, or both in any order.
`decreasing`	How the variables in order.by should be ordered. "freq" is decreasing (greatest to least) and "degree" is increasing (least to greatest)
`show.numbers`	Show numbers of intersection sizes above bars
`number.angles`	The angle of the numbers atop the intersection size bars
`group.by`	How the data should be grouped ("degree" or "sets")

`cutoff`	The number of intersections from each set (to cut off at) when aggregating by sets
`queries`	Unified querie of intersections, elements, and custom row functions. Entered as a list that contains a list of queries. query is the type of query being conducted. params are the parameters of the query (if any). color is the color of the points on the plot that will represent the query. If no color is selected one will be provided automatically. active takes TRUE or FALSE, and if TRUE, it will overlay the bars present with the results from the query. If FALSE a tick mark will indicate the intersection size. See examples section on how to do this.
`query.legend`	Position query legend on top or bottom of UpSet plot
`shade.color`	Color of row shading in matrix
`shade.alpha`	Transparency of shading in matrix
`matrix.dot.alpha`	Transparency of the empty intersections points in the matrix
`empty.intersections`	Additionally display empty sets up to nintersects
`color.pal`	Color palette for attribute plots
`boxplot.summary`	Boxplots representing the distribution of a selected attribute for each intersection. Select attributes by entering a character vector of attribute names (e.g. c("Name1", "Name2")). The maximum number of attributes that can be entered is 2.
`effects.summary`	Forest plot. Select the response variable by entering a character attribute name (e.g. "y" or c("survtime", "cens")
`outcome.type`	One of "continuous", "binary", or "survival" to determine the model to implement
`attribute.plots`	Create custom ggplot using intersection data represented in the main bar plot. Prior to adding custom plots, the UpSet plot is set up in a 100 by 100 grid. The attribute.plots parameter takes a list that contains the number of rows that should be allocated for the custom plot, and a list of plots with specified positions. nrows is the number of rows the custom plots should take up. There is already 100 allocated for the custom plot. plots takes a list that contains a function that returns a custom ggplots and the x and y aesthetics for the function. ncols is the number of columns that your ggplots should take up. See examples for how to add custom ggplots.
`scale.intersections`	The scale to be used for the intersection sizes. Options: "identity", "log10", "log2"
`scale.sets`	The scale to be used for the set sizes. Options: "identity", "log10", "log2"
`text.scale`	Numeric, value to scale the text sizes, applies to all axis labels, tick labels, and numbers above bar plot. Can be a universal scale, or a vector containing individual scales in the following format: c(intersection size title, intersection size tick labels, set size title, set size tick labels, set names, numbers above bars)

`set_size.angles`	Numeric, angle to rotate the set size plot x-axis text
`treatment.var`	A character indicating the name of the treatment variable in the dataset
`min.n`	The minimum number of subjects in a subgroup to be included in the plot
`icon`	One of "dots", "pm", "pm.circle", or "value" which determines the icon to use in the matrix plot
`fill.trt`	A logical indicating whether the bar plot is coloured by treatment.
`transpose`	A logical indicating whether to draw the plot vertically.

Details

Visualization of set data in the layout described by Lex and Gehlenborg in `http://www.nature.com/nmeth/journal/v11/n8/abs/nmeth.3033.html`. The plot is modified here to select a treatment variable, compute the treatment effects and display them along with their confidence intervals in a forest plot-like panel.

Note

Data set must be formatted as described on the orginal UpSet github page: `http://github.com/VCG/upset/wiki`.

References

Lex et al. (2014). UpSet: Visualization of Intersecting Sets IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis 2014), vol 20, pp. 1983-1992, (2014). `http://data.caleydo.org/papers/2014_infovis_upset.pdf`

Lex and Gehlenborg (2014). Points of view: Sets and intersections. Nature Methods 11, 779 (2014). `http://www.nature.com/nmeth/journal/v11/n8/abs/nmeth.3033.html`

See Also

Original UpSet Website: `http://vcg.github.io/upset/about/`

UpSetR github for additional examples: `http://github.com/hms-dbmi/UpSetR`

Examples

```
data(prca)
dat <- prca
vars = data.frame(variable = names(dat), index = 1:length(names(dat)))

## 12. SubgroUpSet -----------------------------------------------------------
prca.upset = data.frame(trt = factor(ifelse(prca$rx == 1, "Experimental", "Control")),
                        bm = 1*(prca$bm == 1),
                        pf = 1*(prca$pf == 1),
                        hx = 1*(prca$hx == 1),
                        stage = 1*(prca$stage == 4),
                        age = 1*(prca$age > 75),
                        wt = 1*(prca$weight > 100),
```

```
                        survtime = prca$survtime,
                        cens = prca$cens==1)
subgroupset(prca.upset,
            order.by = "freq",
            empty.intersections = "on",
            sets = c("bm", 'pf', "hx"),
            text.scale = 1.,
            mb.ratio = c(0.25, 0.50,0.20),
            treatment.var = "trt",
            outcome.type = "survival",
            effects.summary = c("survtime", "cens"),
            query.legend = "top", icon = "pm")

subgroupset(prca.upset,
            order.by = "freq",
            empty.intersections = "on",
            sets = c("bm", 'pf', "hx"),
            text.scale = 1.,
            mb.ratio = c(0.25, 0.50,0.20),
            treatment.var = "trt",
            outcome.type = "survival",
            effects.summary = c("survtime", "cens"),
            query.legend = "top", icon = "pm", transpose = TRUE)
```

Index