Package: rankinPlot (via r-universe)

October 9, 2026

Type Package

Title Convenient Plotting for the Modified Rankin Scale and Other Ordinal Outcome Data

Version 1.2.0

Maintainer Hannah Johns <dr.hannah.johns@gmail.com>

Description Provides convenient tools for visualising ordinal outcome data following conventions within stroke research literature. It currently supports the ``Grotta Bar'' approach pioneered by The National Institute of Neurological Disorders and Stroke rt-PA Stroke Study Group (1995) <doi:10.1056/NEJM199512143332401> and Probability-Probability plots for visualising Desirability of Outcome Ranking (DOOR) scales with large numbers of categories proposed by Johns et al. (2026) <doi:10.1177/17474930261475853>.

License GPL (>=2.0)

Encoding UTF-8

LazyData true

Depends R (>= 3.5)

Imports ggplot2 (>= 4.0), scales (>= 1.2), rlang, grDevices, RColorBrewer, lifecycle

Suggests viridis

Config/roxygen2/version 8.1.0

Repository https://hannahjohns.r-universe.dev

Date/Publication 2026-10-09 08:43:11 UTC

RemoteUrl https://github.com/hannahjohns/rankinplot

RemoteRef HEAD

RemoteSha eae889a7451b3f32617bcca037b7fd288a0876b9

Contents

alteplase 2
grottaBar 3
pp_plot 7
remapcap 12

Index **13**

`alteplase` *Effect of time to treatment on efficacy of rt-PA in treating acute stroke*

Description

A dataset reconstructing reported outcomes after stroke as reported in Lees et. al. (2010) which pooled the results of several studies.

Usage

```
alteplase
```

Format

A data frame with 3669 rows and 3 variables:

time Time interval (minutes) between stroke onset and treatment

treat Type of treatment received

mRS Outcome at 3 months measured using the modified Rankin Scale

References

Lees, K. R., Bluhmki, E., Von Kummer, R., Brott, T. G., Toni, D., Grotta, J. C., Albers, G. W., Kaste, M., Marler, J. R., Hamilton, S. A., Tilley, B. C., Davis, S. M., Donnan, G. A., Hacke, W. (2010). Time to treatment with intravenous alteplase and outcome in stroke: an updated pooled analysis of ECASS, ATLANTIS, NINDS, and EPITHET trials. *The Lancet*, 375(9727), 1695-1703.

grottaBar *grottaBar*

Description

Automates the production of a Grotta Bar using `ggplot()`

Usage

```
grottaBar(
  x,
  groupName,
  scoreName,
  strataName = NULL,
  colorScheme = "Blues",
  colorScheme.reverse = FALSE,
  printNumbers = "count",
  nCol = 1,
  dir = "v",
  width = 0.9,
  textSize = 15,
  numberSize = 5,
  textFace = "plain",
  textColor = NULL,
  lineSize = 0.5,
  lineColor = "black",
  drawLines = TRUE,
  returnData = FALSE,
  ...
)
```

Arguments

`x`	a 2- or 3- dimensional table, returned by the table() function
`groupName`	a character string giving the name of the group variable
`scoreName`	a character string giving outcome labels
`strataName`	a character string giving the strata variable name
`colorScheme`	a character string indicating the colors that should be used by the plot, or a discrete fill scale returned by ggplot2.
`colorScheme.reverse`	A logical indicating if the colour scheme should be reversed.
`printNumbers`	a character string indicating if numbers should be printed for each category.
`nCol`	an integer indicating the number of columns to use for displaying stratified results. Has no effect if no stratification is used.

dir	a character indicating if stratified results should be laid out vertically ("v") or horizontally "h".
width	a number adjusting the width of the lines between bars
textSize	a number indicating the size of text labels
numberSize	a number indicating the size of printed numbers
textFace	a character string indicating font face of printed numbers. Can be "plain", "bold", "italic" or "bold.italic".
textColor	vector of colors for text labels
lineSize	a number indicating the thickness of lines in the plot
lineColor	vector color for lines in the plot
drawLines	boolean indicating if connecting lines should be drawn or not
returnData	a boolean indicating if the data used to create the plot should be returned. For expert users only.
...	additional arguments. Ignored except for colourScheme and textColour which will override their counterpart arguments.

Details

This tool produces a "Grotta" bar chart based on a table of count data. A Grotta bar chart is a common data visualisation tool in stroke research, and is in essence a horizontally stacked proportional bar chart showing the distribution of ordinal outcome data (typically the modified Rankin Scale) across groups, with lines drawn connecting categories across groups.

colorScheme supports all color schemes from ColorBrewer. See https://colorbrewer2.org for more information, or run RColorBrewer::display.brewer.all() to display available options. If the number of categories exceeds what is provided by ColorBrewer, additional values are interpolated as needed.

In addition, setting colorScheme to a ggplot2 discrete scale (e.g. ggplot2::scale_fill_brewer()) allows for a user-specified color scheme using the ggplot2 family of scale_fill_ functions.

The options for printNumbers are:

count The raw counts in the table.

proportion The within-group proportion, rounded to 2 decimal places.

percentage The within-group percentage, rounded to 2 decimal places.

count.percentage The raw count with percentage in parentheses.

none Do not print any numbers.

These options may be abbreviated. "p" is not a valid abbreviation as it matches to multiple options. The minimal abbreviation for "count.percentage" is "c.p"

Value

If returnData **is** FALSE A ggplot object

If returnData **is** TRUE A list containing a ggplot object and the data used to generate it

References

National Institute of Neurological Disorders and Stroke rt-PA Stroke Study Group. "Tissue plasminogen activator for acute ischemic stroke." New England Journal of Medicine 333.24 (1995): 1581-1588.

Rohmann, Jessica L., et al. "Adjusted horizontal stacked bar graphs ("Grotta bars") for consistent presentation of observational stroke study results." European Stroke Journal 8.1 (2023): 370-379.

Forrest, Meghan R., et al. "Use of stacked proportional bar graphs ("Grotta bars") in observational neurology research: a meta-research study." Neurology 104.4 (2025): e210169.

Examples

```
df <- alteplase

x <- table(mRS=df$mRS,
           Group=df$treat,
           Time=df$time)

grottaBar(x,groupName="Group",
          scoreName = "mRS",
          strataName="Time"
)


grottaBar(x,groupName="Time",
          scoreName = "mRS",
          strataName="Group",
          textColor = c(rep("black",4),rep("white",3))
)


x <- table(mRS=df$mRS,
           Group=df$treat)


grottaBar(x,groupName="Group",
          scoreName = "mRS",
          colorScheme = FALSE
)

grottaBar(x,groupName="Group",
          scoreName = "mRS",
          colorScheme = ggplot2::scale_fill_brewer(palette = "Spectral", direction=-1)
)


grottaBar(x,groupName="Group",
          scoreName = "mRS",
          colorScheme = FALSE
)+ ggplot2::scale_fill_brewer(palette = "Spectral", direction=-1)
```

```
grottaBar(x,groupName="Group",
         scoreName = "mRS",
         printNumbers = "count.percentage",
         colorScheme = "Greys"
)

# Example with IPTW weights

 alteplase_confounded <- alteplase
 ps = c("0-90" = 0.7,
        "91-180" = 0.6,
        "181-270" = 0.3,
        "271-360" = 0.7
 )
 set.seed(123)
 alteplase_confounded <- by(alteplase_confounded,alteplase_confounded$time,function(x){

         n_alt <- round(nrow(x)*ps[unique(x$time)])
         n_pbo <- round(nrow(x)*(1-ps[unique(x$time)]))

         this_probs <- prop.table(table(x$mRS,x$treat),margin=2)

        rbind(
                data.frame(treat="Alteplase",
                           mRS = sample(as.numeric(rownames(this_probs)),
                                        size = n_alt, replace=TRUE,
                                        prob=this_probs[,"Alteplase"]),
                           time=unique(x$time)
                           ),
                data.frame(treat="Placebo",
                           mRS = sample(as.numeric(rownames(this_probs)),
                                        size = n_pbo, replace=TRUE,
                                        prob=this_probs[,"Placebo"]),
                           time=unique(x$time)
                           )
        )
})
alteplase_confounded <- do.call("rbind",alteplase_confounded)
alteplase_confounded$treat <- factor(alteplase_confounded$treat,levels=levels(alteplase$treat))

# Raw table, with confounding
x_confounded <- xtabs(~ mRS + treat, data=alteplase_confounded)

# Using IPTW to remove confounding

propensity <- ps[alteplase_confounded$time]
alteplase_confounded$iptw_weight <- ifelse(alteplase_confounded$treat == "Alteplase",
                                           1/propensity,
                                           1/(1-propensity)
                                          )
x_weighted <- xtabs(iptw_weight ~ mRS + treat, data=alteplase_confounded)

grottaBar(x_confounded, groupName = "treat", scoreName = "mRS", printNumbers = "percentage")
```

```
grottaBar(x_weighted, groupName = "treat", scoreName = "mRS",printNumbers = "percentage")

# The original dataset for comparison
x <- xtabs(~ mRS + treat, data=alteplase)
grottaBar(x, groupName = "treat", scoreName = "mRS", printNumbers = "percentage")
```

---

pp_plot *pp_plot*

Description

Creates probability-probability plots for visualizing all-to-all comparisons of ranked data across two groups

Usage

```
pp_plot(x,
        groupName,
        scoreName,
        strataName = NULL,
        reverse.scores = FALSE,
        panel = TRUE,
        panel.nCol = NULL,
        panel.dir = "h",
        polygon = panel,
        polygon.alpha = 0.4,
        polygon.color = "#999999",
        polygon.win.fill = "#71f594",
        polygon.tie.fill = "#e8e156",
        polygon.loss.fill= "#f97194",
        contour = FALSE,
        contour.color = "#555555",
        contour.line.color = contour.color,
        contour.label.color = contour.color,
        confint = TRUE,
        confint.angle = "fixed",
        confint.level = 0.95,
        bar = TRUE,
        bar.colorScheme = "Blues",
        bar.colorScheme.reverse = FALSE,
        bar.width = 0.1,
        bar.lineColor = "black",
        bar.linewidth =  0.5,
        bar.text = "count",
        bar.text.size = 5,
        bar.text.color = NULL,
```

```
bar.text.face = "plain",
neutral.color = "#222222",
neutral.linetype = "dashed",
neutral.linewidth = 0.5,
line.color = NULL,
line.linetype = "solid",
strata.text.size = bar.text.size,
...
)
```

Arguments

x	a 2- or 3- dimensional table, returned by the table() function
groupName	a character string giving the name of the group variable
scoreName	a character string giving outcome labels
strataName	a character string giving the strata variable name
reverse.scores	a logical indicating if the order of the scores should be reversed on the plot
panel	a logical indicating if strata should be separated across panels. If true, returns a faceted plot. If false, all strata are condensed into a single panel.
panel.nCol	an integer indicating the number of columns to use for displaying stratified results. Has no effect if no stratification is used or panel is false.
panel.dir	a character indicating if stratified results should be laid out vertically ("v") or horizontally "h". Has no effect if no stratification is used or panel is false.
polygon	A logical indicating if polygons should be drawn to show the proportion of pairs that are wins, losses or ties. Cannot be used if there are strata and panel is false.
polygon.alpha	A numeric value for the transparency (alpha) for the polygons.
polygon.color	A character string giving the border color for the polygons.
polygon.win.fill	A character string giving the fill color for the polygons indicating a region of wins.
polygon.tie.fill	A character string giving the fill colour for the polygons indicating a region of tied pairs.
polygon.loss.fill	A character string giving the fill colour for the polygons indicating a region of losses.
contour	A logical indicating if contours should be drawn indicating where the probability-probability plot should sit if the proportional odds assumption is met.
contour.color	A character string giving the color of the contours
contour.line.color	A character string giving the color of the contour lines
contour.label.color	A character string giving the color of the contour text

`confint` A logical indicating if confidence intervals representing should be drawn around each point. See details.

`confint.angle` A character string indicating the direction to draw the angle. See details.

`confint.level` A numeric value indicating the level of confidence for the confidence interval.

`bar` A logical indicating if bars should be drawn to indicate the distribution of the outcome in each group and strata.

`bar.colorScheme`
A character string indicating the colour scheme to use for the bars. See details.

`bar.colorScheme.reverse`
A logical indicating if the colour scheme should be reversed.

`bar.width` A numeric value indicating the width of the bars

`bar.lineColor` A character string indicating the colour of the bar borders.

`bar.linewidth` A numeric value indicating the width of the bar borders.

`bar.text` a character string indicating if numbers should be printed for each category.

`bar.text.size` a number indicating the size of text labels

`bar.text.color` A vector of colors for text labels

`bar.text.face` A character string indicating font face of printed numbers. Can be "plain", "bold", "italic" or "bold.italic".

`neutral.color` A character string indicating the color of the neutral line

`neutral.linetype`
A character string indicating the line type of the neutral line

`neutral.linewidth`
A numeric value indicating the width of the neutral line

`line.color` A character string indicating the colour to draw the probability-probability line with, or a discrete color scale returned by ggplot2 to have this vary by strata.

`line.linetype` A character string indicating the linetype to draw the probability-probability line with, or a discrete linetype scale returned by ggplot2 to have this vary by strata.

`strata.text.size`
A number indicating the size of the text to draw the strata label. Only relevant if data is stratified and panel is false.

`...` Any other arguments. Ignored, but used to catch e.g. British spelling of "color".

Details

Confidence intervals are estimated using Fisher's exact test and correspond to an odds ratio. They are visually represented as segments on the figure, where the start and end points correspond to cumulative probabilities that match the upper/lower bound of the odds ratio. Two options are given for the direction of these lines. "fixed" indicates that all confidence intervals should be drawn at 45 degrees, while "proportional.odds" indicates that they should be drawn perpendicular to the proportional odds contour line.

`bar.colorScheme` supports all color schemes from ColorBrewer. See https://colorbrewer2.org for more information, or run `RColorBrewer::display.brewer.all()` to display available options.

If the number of categories exceeds what is provided by ColorBrewer, additional values are interpolated as needed.

In addition, setting colorScheme to a ggplot2 discrete scale (e.g. `ggplot2::scale_fill_brewer()` allows for a user-specified color scheme using the ggplot2 family of `scale_fill_` functions.

The options for `bar.text` are:

`"count"` The raw counts in the table.

`"proportion"` The within-group proportion, rounded to 2 decimal places.

`"percentage"` The within-group percentage, rounded to 2 decimal places.

`"count.percentage"` The raw count with percentage in parentheses.

`"none"`, `FALSE`, `NULL` or `NA` Do not print any numbers.

These options may be abbreviated. `"p"` is not a valid abbreviation as it matches to multiple options. The minimal abbreviation for `"count.percentage"` is `"c.p"`

Value

A ggplot object containing the plot.

References

Johns, Hannah, et al. "Practical guidance for Win Statistics and Tournament Methods for multi-faceted outcomes in stroke research: Review and recommendations." International Journal of Stroke (2026) DOI: https://doi.org/10.1177/17474930261475853

Examples

```
df <- alteplase

x <- table(mRS=df$mRS,
           Group=df$treat,
           Time=df$time)

pp_plot(x,
       groupName =  "Group",
       scoreName = "mRS",
       strataName = "Time",
       panel = TRUE,
       confint = TRUE
)


pp_plot(x,
       groupName =  "Group",
       scoreName = "mRS",
       strataName = "Time",
       panel = FALSE,
       confint = FALSE
)
```

```
df <- remapcap
df <- df[which(df$group %in% c("uc","shock")),]
df$group <-droplevels(df$group)

x <- table(Score=df$score,
          Group=df$group
)



pp_plot(x,groupName="Group",
       scoreName = "Score",
       reverse.scores = TRUE,
       bar.text = FALSE,
       confint.angle = "fixed",
       bar.colorScheme = "RdBu",
       bar.colorScheme.reverse = TRUE
)

# Visually inspect proportional odds assumption
pp_plot(x,groupName="Group",
       scoreName = "Score",
       reverse.scores = TRUE,
       bar.text = FALSE,
       confint.angle = "proportional.odds",
       bar.colorScheme = "RdBu",
       bar.colorScheme.reverse = TRUE,
       contour = TRUE,
       polygon = FALSE,
       neutral.color = "darkred", neutral.linewidth =1
)



# Transform the data for comparison of two arms against a common control arm

df_shock <- remapcap[which(remapcap$group %in% c("uc","shock")),]
df_fixed <- remapcap[which(remapcap$group %in% c("uc","fixed")),]

df_shock$group2 <- ifelse(df_shock$group=="uc","Usual Care","Intervention")
df_fixed$group2 <- ifelse(df_fixed$group=="uc","Usual Care","Intervention")
df_shock$intervention <- "Shock"
df_fixed$intervention <- "Fixed"

df <- rbind(df_shock,df_fixed)
df$group2 <- factor(df$group2,levels=c("Usual Care","Intervention"))


x <- table(Score=df$score,
          Group=df$group2,
          Intervention=df$intervention
)
```

```
pp_plot(x,groupName="Group",
       scoreName = "Score",
       strataName = "Intervention",
       reverse.scores = TRUE,
       bar.text = FALSE,
       confint= FALSE,
       panel=FALSE,
       bar.colorScheme = "RdBu",
       bar.colorScheme.reverse = TRUE
)
```

`remapcap` *Effect of hydrocortisone on mortality and organ support in patients with severe COVID-19*

Description

A dataset reconstructing reported outcomes in severe COVID-19 as reported in Angus et al. (2020)

Usage

```
remapcap
```

Format

A data frame with 379 rows and 6 variables:

group The treatment group in the trial. uc = Usual Care, shock = Shock-dependent hydrocortisone, fixed = Fixed-dose hydrocortisone

deathTime Time of death, censored at 21 days

dischargeTime Time of ICU discharge, censored at 21 days

deathStatus If death was observed (0=no, 1=yes)

dischargeStatus If ICU discharde was observed (0=no, 1=yes)

score An ordinal score ranging from -1 to 21 covering death and organ support-free days

References

Angus, Derek C., et al. "Effect of hydrocortisone on mortality and organ support in patients with severe COVID-19: the REMAP-CAP COVID-19 corticosteroid domain randomized clinical trial." Jama 324.13 (2020): 1317-1329.

Index

∗ **datasets**
alteplase, 2
remapcap, 12

alteplase, 2

display.brewer.all, *4*

grottaBar, 3

pp_plot, 7

remapcap, 12

scale_fill_brewer, *4*