Package ‘plotROC’

July 23, 2025

Type Package

Title Generate Useful ROC Curve Charts for Print and Interactive Use

Version 2.3.1

Date 2023-10-05

Maintainer Michael C. Sachs <`sachsmc@gmail.com`>

Description Most ROC curve plots obscure the cutoff values and inhibit interpretation and comparison of multiple curves. This attempts to address those shortcomings by providing plotting and interactive tools. Functions are provided to generate an interactive ROC curve plot for web use, and print versions. A Shiny application implementing the functions is also included.

License MIT + file LICENSE

URL `https://sachsmc.github.io/plotROC/`

BugReports `https://github.com/sachsmc/plotROC/issues/`

Depends R (>= 3.0.0), ggplot2

Imports methods, grid, gridSVG, shiny, plyr, rlang

Suggests knitr, testthat, stringr, survivalROC, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.2.3

NeedsCompilation no

Author Michael C. Sachs [aut, cre],
Robert W. Corty [ctb],
Luis Crouch [ctb] (Modification to calc_auc)

Repository CRAN

Date/Publication 2023-10-06 12:40:02 UTC

Contents

`calculate_multi_roc` *Calculate the Empirical ROC curves for multiple biomarkers*

Description

Deprecated, use geom_roc instead

Usage

```
calculate_multi_roc(data, M_string, D_string)
```

Arguments

`data`	data frame containing at least 1 marker and the common class labels, coded as 0 and 1
`M_string`	vector of marker column names
`D_string`	class label column name

Value

List of data frames containing cutoffs, and estimated true and false positive fractions

---

`calculate_roc` *Calculate the Empirical ROC curve*

Description

Deprecated, use geom_roc instead

Usage

```
calculate_roc(M, D, ci = FALSE, alpha = 0.05)
```

Arguments

`M`	continuous marker values or predictions of class labels
`D`	class labels, must be coded as 0 and 1. If not numeric with 0/1, then plotROC assumes the first level in sort order is healthy status, with a warning.
`ci`	Logical, if true, will calculate exact joint confidence regions for the TPF and FPF
`alpha`	Confidence level, ignored if `ci = FALSE`

Details

Confidence intervals for TPF and FPF are calculated using the exact method of Clopper and Pearson (1934) each at the level `1 - sqrt(1 - alpha)`. Based on result 2.4 from Pepe (2003), the cross-product of these intervals yields a 1 - alpha

Value

A dataframe containing cutoffs, estimated true and false positive fractions, and confidence intervals if `ci = TRUE`.

`calc_auc` *Calculate the Area under the ROC curve*

Description

Given a ggplot object with a GeomRoc layer, computes the area under the ROC curve for each group

Usage

```
calc_auc(ggroc)
```

Arguments

`ggroc`	A ggplot object that contains a GeomRoc layer

Value

A data frame with the estimated AUCs for each panel and group

Examples

```
D.ex <- rbinom(50, 1, .5)
rocdata <- data.frame(D = c(D.ex, D.ex),
                     M = c(rnorm(50, mean = D.ex, sd = .4), rnorm(50, mean = D.ex, sd = 1)),
                       Z = c(rep("A", 50), rep("B", 50)))

ggroc <- ggplot(rocdata, aes(m = M, d = D)) + geom_roc()
calc_auc(ggroc)
ggroc2 <- ggplot(rocdata, aes(m = M, d = D, color = Z)) + geom_roc()
calc_auc(ggroc2)
```

`direct_label` *Add direct labels to a ROC plot*

Description

Add direct labels to a ROC plot

Usage

```
direct_label(
  ggroc_p,
  labels = NULL,
  label.angle = 45,
  nudge_x = 0,
  nudge_y = 0,
  size = 6,
  ...
)
```

Arguments

`ggroc_p`	A ggplot object that contains a geom_roc layer
`labels`	vector of labels to add directly to the plot next to the curves. If multiple curves, must be in the same order as the grouping factor. If NULL, attempts to determine labels from the ggroc_p object
`label.angle`	angle of adjustment for the direct labels
`nudge_x, nudge_y`	Horizontal and vertical adjustment to nudge labels by. These can be scalars or vectors the same length as the number of labels
`size`	Size of labels
`...`	Other arguments passed to annotate

`export_interactive_roc`

Generate svg code for an ROC curve object

Description

Takes a ggplot object that contains a GeomRoc layer and returns a string that contains html suitable for creating a standalone interactive ROC curve plot.

Usage

```
export_interactive_roc(
  ggroc_p,
  add.cis = TRUE,
  hide.points = FALSE,
  prefix = "a",
  width = 6,
  height = 6,
  omit.js = FALSE,
  style = style_roc(theme = theme_grey()),
  ...
)
```

Arguments

`ggroc_p`	A ggplot object with a GeomRoc layer and optionally a GeomRocci layer as returned by geom_roc and/or geom_rocci. It can be modified with annotations, themes, etc.
`add.cis`	Logical, if true, removes the current confidence interval layer (if present) and replaces it with a denser layer of confidence regions
`hide.points`	Logical, if true, hides points layer so that points with cutoff values are only visible when hovering. Recommended for plots containing more than 3 curves.
`prefix`	A string to assign to the objects within the svg. Enables unique identification by the javascript code
`width, height`	Width and height in inches of plot
`omit.js`	Logical. If true, omit inclusion of javascript source in output. Useful for documents with multiple interactive plots
`style`	A call to the function style_roc
`...`	Other arguments passed to geom_rocci when `add.cis` = `TRUE`

Details

If you intend to include more than one of these objects in a single page, use a different `prefix` string for each one. To use this function in knitr, use the chunk options `fig.keep='none'` and `results = 'asis'`, then `cat()` the resulting string to the output. See the vignette for examples. Older browsers (< IE7) are not supported.

Value

A character object containing the html necessary to plot the ROC curve in a web browser

---

GeomRoc *Empirical Receiver Operating Characteristic Curve*

Description

Display the empirical ROC curve. Useful for characterizing the classification accuracy of continuous measurements for predicting binary states

Usage

```
GeomRoc

geom_roc(
  mapping = NULL,
  data = NULL,
  stat = "roc",
  n.cuts = 10,
  arrow = NULL,
  lineend = "butt",
  linejoin = "round",
  linemitre = 1,
  linealpha = 1,
  pointalpha = 1,
  pointsize = 0.5,
  labels = TRUE,
  labelsize = 3.88,
  labelround = 1,
  na.rm = TRUE,
  cutoffs.at = NULL,
  cutoff.labels = NULL,
  position = "identity",
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)
```

Arguments

`mapping`	Set of aesthetic mappings created by `aes()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.

`data`	The data to be displayed in this layer. There are three options: If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`. A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created. A `function` will be called with a single argument, the plot data. The return value must be a `data.frame`, and will be used as the layer data. A `function` can be created from a `formula` (e.g. `~ head(.x, 10)`).
`stat`	Use to override the default connection between `geom_roc` and `stat_roc`.
`n.cuts`	Number of cutpoints to display along each curve
`arrow`	Arrow specification, as created by `arrow`
`lineend`	Line end style (round, butt, square)
`linejoin`	Line join style (round, mitre, bevel)
`linemitre`	Line mitre limit (number greater than 1)
`linealpha`	Alpha level for the lines, alpha.line is deprecated
`pointalpha`	Alpha level for the cutoff points, alpha.point is deprecated
`pointsize`	Size of cutoff points, size.point is deprecated
`labels`	Logical, display cutoff text labels
`labelsize`	Size of cutoff text labels
`labelround`	Integer, number of significant digits to round cutoff labels
`na.rm`	Remove missing values from curve
`cutoffs.at`	Vector of user supplied cutoffs to plot as points. If non-NULL, it will override the values of n.cuts and plot the observed cutoffs closest to the user-supplied ones.
`cutoff.labels`	vector of user-supplied labels for the cutoffs. Must be a character vector of the same length as cutoffs.at.
`position`	Position adjustment, either as a string naming the adjustment (e.g. `"jitter"` to use `position_jitter`), or the result of a call to a position adjustment function. Use the latter if you need to change the settings of the adjustment.
`show.legend`	logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display.
`inherit.aes`	If `FALSE`, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. `borders()`.
`...`	Other arguments passed on to `layer()`. These are often aesthetics, used to set an aesthetic to a fixed value, like `colour = "red"` or `size = 3`. They may also be parameters to the paired geom/stat.

Format

An object of class `GeomRoc` (inherits from `Geom`, `ggproto`, `gg`) of length 6.

Computed variables

false_positive_fraction estimate of false positive fraction

true_positive_fraction estimate of true positive fraction

cutoffs values of m at which estimates are calculated

Aesthetics

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

- `x` The FPF estimate. This is automatically mapped by stat_roc
- `y` The TPF estimate. This is automatically mapped by stat_roc smallest level in sort order is assumed to be 0, with a warning
- `alpha`
- `color`
- `fill`
- `linetype`
- `size`

See Also

See `geom_rocci` for displaying rectangular confidence regions for the empirical ROC curve, `style_roc` for adding guidelines and labels, and `direct_label` for adding direct labels to the curves. Also export_interactive_roc for creating interactive ROC curve plots for use in a web browser.

Examples

```
D.ex <- rbinom(50, 1, .5)
rocdata <- data.frame(D = c(D.ex, D.ex),
                  M = c(rnorm(50, mean = D.ex, sd = .4), rnorm(50, mean = D.ex, sd = 1)),
                   Z = c(rep("A", 50), rep("B", 50)))

ggplot(rocdata, aes(m = M, d = D)) + geom_roc()

ggplot(rocdata, aes(m = M, d = D, color = Z)) + geom_roc()
ggplot(rocdata, aes(m = M, d = D)) + geom_roc() + facet_wrap(~ Z)
ggplot(rocdata, aes(m = M, d = D)) + geom_roc(n.cuts = 20)
ggplot(rocdata, aes(m = M, d = D)) + geom_roc(cutoffs.at = c(1.5, 1, .5, 0, -.5))
ggplot(rocdata, aes(m = M, d = D)) + geom_roc(labels = FALSE)
ggplot(rocdata, aes(m = M, d = D)) + geom_roc(size = 1.25)
```

`geom_rocci` *Confidence regions for the ROC curve*

Description

Display rectangular confidence regions for the empirical ROC curve.

Usage

```
geom_rocci(
  mapping = NULL,
  data = NULL,
  stat = "rocci",
  ci.at = NULL,
  sig.level = 0.05,
  na.rm = TRUE,
  alpha.box = 0.3,
  labels = TRUE,
  labelsize = 3.88,
  labelround = 1,
  position = "identity",
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)

GeomRocci
```

Arguments

`mapping`	Set of aesthetic mappings created by `aes()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.
`data`	The data to be displayed in this layer. There are three options: If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`. A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created. A `function` will be called with a single argument, the plot data. The return value must be a `data.frame`, and will be used as the layer data. A `function` can be created from a `formula` (e.g. `~ head(.x, 10)`).
`stat`	Use to override the default connection between `geom_rocci` and `stat_rocci`.
`ci.at`	Vector of values in the range of the biomarker where confidence regions will be displayed

`sig.level`	Significance level for the confidence regions
`na.rm`	If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed.
`alpha.box`	Alpha level for the confidence regions
`labels`	If TRUE, adds text labels for the cutoffs where the confidence regions are displayed
`labelsize`	Size of cutoff text labels
`labelround`	Integer, number of significant digits to round cutoff labels
`position`	Position adjustment, either as a string naming the adjustment (e.g. `"jitter"` to use `position_jitter`), or the result of a call to a position adjustment function. Use the latter if you need to change the settings of the adjustment.
`show.legend`	logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display.
`inherit.aes`	If `FALSE`, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. `borders()`.
`...`	Other arguments passed on to `layer()`. These are often aesthetics, used to set an aesthetic to a fixed value, like `colour = "red"` or `size = 3`. They may also be parameters to the paired geom/stat.

Format

An object of class `GeomRocci` (inherits from `Geom`, `ggproto`, `gg`) of length 6.

Aesthetics

`geom_rocci` understands the following aesthetics (required aesthetics are in bold). `stat_rocci` automatically maps the estimates to the required aesthetics:

- `x` The FPF estimate
- `y` The TPF estimate
- `xmin` Lower confidence limit for the FPF
- `xmax` Upper confidence limit for the FPF
- `ymin` Lower confidence limit for the TPF
- `ymax` Upper confidence limit for the TPF
- `alpha`
- `color`
- `fill`
- `linetype`
- `size`

See Also

See `geom_roc` for the empirical ROC curve, `style_roc` for adding guidelines and labels, and `direct_label` for adding direct labels to the curves. Also export_interactive_roc for creating interactive ROC curve plots for use in a web browser.

Examples

```
D.ex <- rbinom(50, 1, .5)
rocdata <- data.frame(D = c(D.ex, D.ex),
                M = c(rnorm(50, mean = D.ex, sd = .4), rnorm(50, mean = D.ex, sd = 1)),
                 Z = c(rep("A", 50), rep("B", 50)))

ggplot(rocdata, aes(m = M, d = D)) + geom_roc() + geom_rocci()
ggplot(rocdata, aes(m = M, d = D, color = Z)) + geom_roc() + geom_rocci()
ggplot(rocdata, aes(m = M, d = D, color = Z)) + geom_roc() + geom_rocci(sig.level = .01)
ggplot(rocdata, aes(m = M, d = D)) + geom_roc(n.cuts = 0) +
geom_rocci(ci.at = quantile(rocdata$M, c(.1, .25, .5, .75, .9)))
ggplot(rocdata, aes(m = M, d = D, color = Z)) + geom_roc() + geom_rocci(linetype = 1)
```

`getD3` *Reads included JavaScript functions and returns them as a string for pasting into a webpage*

Description

Reads included JavaScript functions and returns them as a string for pasting into a webpage

Usage

```
getD3()
```

`ggroc` *Plot an ROC curve*

Description

Deprecated, use geom_roc instead

Usage

```
ggroc(
  rocdata,
  fpf_string = "FPF",
  tpf_string = "TPF",
  c_string = "c",
  ci = FALSE,
```

```
  label = NULL,
  label.adj.x = 0,
  label.adj.y = 0,
  label.angle = 45,
  plotmath = FALSE,
  xlabel = "False positive fraction",
  ylabel = "True positive fraction"
)
```

Arguments

`rocdata`	Data frame containing true and false positive fractions, and cutoff values
`fpf_string`	Column name identifying false positive fraction column
`tpf_string`	Column name identifying true positive fraction column
`c_string`	Column name identifying cutoff values
`ci`	Logical, not supported
`label`	Not supported
`label.adj.x`	Not supported
`label.adj.y`	Not supported
`label.angle`	Not supported
`plotmath`	Not supported
`xlabel`	Defaults to "False positive fraction"
`ylabel`	Defaults to "True positive fraction"

Value

A ggplot object

`melt_roc` *Transform biomarkers stored as wide to long*

Description

Multiple biomarkers measured on the same subjects are often stored as multiple columns in a data frame. This is a convenience function that transforms the data into long format, suitable for use with ggplot and geom_roc

Usage

```
melt_roc(data, d, m, names = NULL)
```

Arguments

`data`	Data frame containing disease status and biomarkers stored in columns
`d`	Column containing binary disease status. Can be a column name or index
`m`	Vector of column names or indices identifying biomarkers
`names`	Optional vector of names to assign to the biomarkers. If NULL, names will be taken from the column names

Value

A data frame in long format with three columns: D = binary disease status, M = biomarker value, and name = biomarker name

Examples

```
D.ex <- rbinom(50, 1, .5)
widedata <- data.frame(D = D.ex, M1 = rnorm(50, mean = D.ex, sd = 1),
   M2 = rnorm(50, mean = D.ex, sd = .5))
longdata <- melt_roc(widedata, "D", c("M1", "M2"))
ggplot(longdata, aes(d = D, m = M, color = name)) + geom_roc()
```

`multi_ggroc` *Plot multiple ROC curves*

Description

Given a list of results computed by calculate_roc, plot the curve using ggplot with sensible defaults. Pass the resulting object and data to export_interactive_roc, plot_interactive_roc, or plot_journal_roc.

Usage

```
multi_ggroc(
  datalist,
  fpf_string = rep("FPF", length(datalist)),
  tpf_string = rep("TPF", length(datalist)),
  c_string = rep("c", length(datalist)),
  label = NULL,
  legend = TRUE,
  label.adj.x = rep(0, length(datalist)),
  label.adj.y = rep(0, length(datalist)),
  label.angle = rep(45, length(datalist)),
  plotmath = FALSE,
  xlabel = "False positive fraction",
  ylabel = "True positive fraction"
)
```

Arguments

`datalist`	List of data frames each containing true and false positive fractions and cutoffs
`fpf_string`	Column names identifying false positive fraction
`tpf_string`	Column names identifying true positive fraction
`c_string`	Column names identifying cutoff values
`label`	Not supported.
`legend`	If true, draws legend
`label.adj.x`	Not supported.
`label.adj.y`	Not supported.
`label.angle`	Not supported.
`plotmath`	Logical. Not supported.
`xlabel`	Defaults to "False positive fraction"
`ylabel`	Defaults to "True positive fraction"

Value

A ggplot object

`plotROC` *Tools for plotting ROC Curves*

Description

Generate Useful ROC Curve Charts for Print and Interactive Use. This defines a set of stats and geoms for use with ggplot2. In addition, ggplot objects created with these geoms can be exported and turned into interactive plots for use on the web. The interactive features include hovering to display hidden labels, and clicking to reveal confidence regions.

Details

To get started, see geom_roc, geom_rocci, or the examples below. For transforming data, we also provide the convenience function melt_roc.

The vignette contains some examples, along with explanations of the results. To view, run `vignette("examples", package = "plotROC")`

Author(s)

Michael Sachs (@sachsmc)

See Also

Useful links:

- `https://sachsmc.github.io/plotROC/`
- Report bugs at `https://github.com/sachsmc/plotROC/issues/`

Examples

```
D.ex <- rbinom(50, 1, .5)
rocdata <- data.frame(D = c(D.ex, D.ex),
                 M = c(rnorm(50, mean = D.ex, sd = .4), rnorm(50, mean = D.ex, sd = 1)),
                  Z = c(rep("A", 50), rep("B", 50)))

ggplot(rocdata, aes(m = M, d = D)) + geom_roc() + geom_rocci()

ggplot(rocdata, aes(m = M, d = D)) + geom_roc() + style_roc()
ggplot(rocdata, aes(m = M, d = D, color = Z)) + geom_roc() + geom_rocci()
ggplot(rocdata, aes(m = M, d = D, color = Z)) + geom_roc() + geom_rocci(sig.level = .01)
ggplot(rocdata, aes(m = M, d = D)) + geom_roc(n.cuts = 0) +
geom_rocci(ci.at = quantile(rocdata$M, c(.1, .25, .5, .75, .9)))
ggplot(rocdata, aes(m = M, d = D, color = Z)) + geom_roc() + geom_rocci(linetype = 1)

rocplot <- ggplot(rocdata, aes(m = M, d = D)) + geom_roc()

plot_interactive_roc(rocplot)
plot_interactive_roc(rocplot + aes(color = Z))
plot_interactive_roc(rocplot + facet_wrap( ~ Z))
```

`plot_interactive_roc` *Generate a standalone html document displaying an interactive ROC curve*

Description

Generate a standalone html document displaying an interactive ROC curve

Usage

```
plot_interactive_roc(ggroc, file = NULL, ...)
```

Arguments

`ggroc`	An object as returned by ggroc or multi_ggroc. It can be modified with annotations, themes, etc.
`file`	A path to save the result to. If NULL, will save to a temporary directory
`...`	arguments passed to export_interactive_roc

Value

NULL opens an interactive document in Rstudio or the default web browser

---

`plot_journal_roc` *Plot an ROC curve for use in print*

Description

Deprecated, use style_roc instead

Usage

```
plot_journal_roc(
  ggroc_p,
  font.size = 3,
  n.cuts = 20,
  ci.at = NULL,
  opacity = 0.3,
  lty = NULL,
  color = NULL,
  lwd = NULL,
  legend = FALSE
)
```

Arguments

`ggroc_p`	An object as returned by ggroc or multi_ggroc. It can be modified with annotations, themes, etc.
`font.size`	Not supported
`n.cuts`	Not supported
`ci.at`	Not supported
`opacity`	Not supported
`lty`	Not supported
`color`	Not supported
`lwd`	Not supported
`legend`	Not supported

Value

A ggplot object

`roc_key` *Key for ROC geom*

Description

Key for ROC geom

Usage

```
roc_key(data, params, size)
```

Arguments

`data`	Data created by stat
`params`	parameters
`size`	Size

`shiny_plotROC` *Start the plotROC Shiny app*

Description

A convenience function to easily start the shiny application. It will open in Rstudio, or in the default web browser.

Usage

```
shiny_plotROC()
```

`StatRoc` *Calculate the empirical Receiver Operating Characteristic curve*

Description

Given a binary outcome d and continuous measurement m, computes the empirical ROC curve for assessing the classification accuracy of m

Usage

```
StatRoc

stat_roc(
  mapping = NULL,
  data = NULL,
  geom = "roc",
  position = "identity",
  show.legend = NA,
  inherit.aes = TRUE,
  na.rm = TRUE,
  max.num.points = 1000,
  increasing = TRUE,
  ...
)
```

Arguments

`mapping`	Set of aesthetic mappings created by `aes()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.
`data`	The data to be displayed in this layer. There are three options: If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`. A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created. A `function` will be called with a single argument, the plot data. The return value must be a `data.frame`, and will be used as the layer data. A `function` can be created from a `formula` (e.g. `~ head(.x, 10)`).
`geom`	The geometric object to use to display the data, either as a `ggproto` `Geom` subclass or as a string naming the geom stripped of the `geom_` prefix (e.g. `"point"` rather than `"geom_point"`)
`position`	Position adjustment, either as a string naming the adjustment (e.g. `"jitter"` to use `position_jitter`), or the result of a call to a position adjustment function. Use the latter if you need to change the settings of the adjustment.
`show.legend`	logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display.
`inherit.aes`	If `FALSE`, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. `borders()`.
`na.rm`	Remove missing observations
`max.num.points`	maximum number of points to plot
`increasing`	TRUE (default) if M is positively associated with Pr(D = 1), if FALSE, assumes M is negatively associated with Pr(D = 1)

`...`	Other arguments passed on to `layer()`. These are often aesthetics, used to set an aesthetic to a fixed value, like `colour = "red"` or `size = 3`. They may also be parameters to the paired geom/stat.

Format

An object of class `StatRoc` (inherits from `Stat`, `ggproto`, `gg`) of length 6.

Aesthetics

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

- `m` The continuous biomarker/predictor
- `d` The binary outcome, if not coded as 0/1, the smallest level in sort order is assumed to be 0, with a warning
- `alpha` Controls the label alpha, see also `linealpha` and `pointalpha`
- `color`
- `linetype`
- `size` Controls the line weight, see also `pointsize` and `labelsize`

Computed variables

false_positive_fraction estimate of false positive fraction

true_positive_fraction estimate of true positive fraction

cutoffs values of m at which estimates are calculated

Examples

```
D.ex <- rbinom(50, 1, .5)
rocdata <- data.frame(D = c(D.ex, D.ex),
                   M = c(rnorm(50, mean = D.ex, sd = .4), rnorm(50, mean = D.ex, sd = 1)),
                    Z = c(rep("A", 50), rep("B", 50)))

ggplot(rocdata, aes(m = M, d = D)) + stat_roc()
```

`StatRocci` *Calculate confidence regions for the empirical ROC curve*

Description

Confidence intervals for TPF and FPF are calculated using the exact method of Clopper and Pearson (1934) each at the level `1 - sqrt(1 - alpha)`. Based on result 2.4 from Pepe (2003), the cross-product of these intervals yields a 1 - alpha

Usage

```
StatRocci

stat_rocci(
  mapping = NULL,
  data = NULL,
  geom = "rocci",
  position = "identity",
  show.legend = NA,
  inherit.aes = TRUE,
  ci.at = NULL,
  sig.level = 0.05,
  na.rm = TRUE,
  ...
)
```

Arguments

`mapping`	Set of aesthetic mappings created by `aes()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.
`data`	The data to be displayed in this layer. There are three options: If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`. A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created. A `function` will be called with a single argument, the plot data. The return value must be a `data.frame`, and will be used as the layer data. A `function` can be created from a `formula` (e.g. `~ head(.x, 10)`).
`geom`	The geometric object to use to display the data, either as a `ggproto` `Geom` subclass or as a string naming the geom stripped of the `geom_` prefix (e.g. `"point"` rather than `"geom_point"`)
`position`	Position adjustment, either as a string naming the adjustment (e.g. `"jitter"` to use `position_jitter`), or the result of a call to a position adjustment function. Use the latter if you need to change the settings of the adjustment.
`show.legend`	logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display.
`inherit.aes`	If `FALSE`, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. `borders()`.
`ci.at`	Vector of cutoffs at which to display confidence regions. If NULL, will automatically choose 3 evenly spaced points to display the regions
`sig.level`	Significance level for the confidence regions

`na.rm`	Remove missing observations
`...`	Other arguments passed on to `layer()`. These are often aesthetics, used to set an aesthetic to a fixed value, like `colour = "red"` or `size = 3`. They may also be parameters to the paired geom/stat.

Format

An object of class `StatRocci` (inherits from `Stat`, `ggproto`, `gg`) of length 6.

Aesthetics

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

- `m` The continuous biomarker/predictor
- `d` The binary outcome, if not coded as 0/1, the smallest level in sort order is assumed to be 0, with a warning
- `alpha`
- `color`
- `fill`
- `linetype`
- `size`

Computed variables

FPF estimate of false positive fraction

TPF estimate of true positive fraction

cutoffs values of m at which estimates are calculated

FPFL lower bound of confidence region for FPF

FPFU upper bound of confidence region for FPF

TPFL lower bound of confidence region for TPF

TPFU upper bound of confidence region for TPF

References

- Clopper, C. J., and Egon S. Pearson. "The use of confidence or fiducial limits illustrated in the case of the binomial." Biometrika (1934): 404-413.
- Pepe, M.S. "The Statistical Evaluation of Medical Tests for Classification and Prediction." Oxford (2003).

Examples

```
D.ex <- rbinom(50, 1, .5)
rocdata <- data.frame(D = c(D.ex, D.ex),
                 M = c(rnorm(50, mean = D.ex, sd = .4), rnorm(50, mean = D.ex, sd = 1)),
                  Z = c(rep("A", 50), rep("B", 50)))

ggplot(rocdata, aes(m = M, d = D)) + geom_roc() + stat_rocci()
ggplot(rocdata, aes(m = M, d = D)) + geom_roc() +
stat_rocci(ci.at = quantile(rocdata$M, c(.1, .3, .5, .7, .9)))
```

`style_roc` *Add guides and annotations to a ROC plot*

Description

Adds a diagonal guideline, minor grid lines, and optionally direct labels to ggplot objects containing a geom_roc layer.

Usage

```
style_roc(
  major.breaks = c(0, 0.1, 0.25, 0.5, 0.75, 0.9, 1),
  minor.breaks = c(seq(0, 0.1, by = 0.01), seq(0.9, 1, by = 0.01)),
  guide = TRUE,
  xlab = "False positive fraction",
  ylab = "True positive fraction",
  theme = theme_bw
)
```

Arguments

`major.breaks`	vector of breakpoints for the major gridlines and axes
`minor.breaks`	vector of breakpoints for the minor gridlines and axes
`guide`	logical, if TRUE draws diagonal guideline
`xlab`	X-axis label
`ylab`	Y-axis label
`theme`	Theme function compatible with ggplot2

Examples

```
D.ex <- rbinom(50, 1, .5)
fakedata <- data.frame(M1 = rnorm(50, mean = D.ex),
   D = D.ex)
ggplot(fakedata, aes(m = M1, d = D)) + geom_roc() + style_roc()
ggplot(fakedata, aes(m = M1, d = D)) + geom_roc() + style_roc(xlab = "1 - Specificity")
ggplot(fakedata, aes(m = M1, d = D)) + geom_roc() + style_roc(theme = theme_grey)
```

`verify_d` *Check that D is suitable for using as binary disease status*

Description

Checks for two classes and gives a warning message indicating which level is assumed to be 0/1. Throws an error if more than two levels appear in D.

Usage

```
verify_d(D)
```

Arguments

`D`	Vector that will be checked for 2-class labels

Value

A vector the same length as D that takes values 0, indicating no disease or 1 indicating disease.

Examples

```
verify_d(c(1, 0, 1))
## Not run:
verify_d(c(TRUE, FALSE, TRUE)) #warning
verify_d(c("Dead", "Alive", "Dead")) #warning
verify_d(c("Disease", "Healthy", "Missing")) #error

## End(Not run)
```

Index