Package 'viopoints'

February 20, 2015

Type Package

Title 1-D Scatter Plots with Jitter Using Kernel Density Estimates

Version 0.2-1

Date 2011-08-28

Author Yuki Atoh

Maintainer Yuki Atoh <yuki.atoh@gmail.com>

Description viopoints draws one dimensional scatter plots with jitter using kernel density estimates in a similar way to violin plots.

License GPL (>= 2)

LazyLoad yes

Repository CRAN

Date/Publication 2011-08-28 05:38:18

NeedsCompilation no

R topics documented:

`viopoints` *1-D Scatter Plots with Jitter Using Kernel Density Estimates*

Description

`viopoints` draws one dimensional scatter plots with jitter using kernel density estimates. The plots are similar to `stripchart` when the sample size is small. If the sample size is large, the plots are similar to violin plots.

Usage

```
viopoints(x, ...)

## S3 method for class 'formula'
viopoints(formula, data = NULL, ..., subset, na.action = NULL)

## Default S3 method:
viopoints(x, ..., groups, na.group = FALSE,
    method = "violin", side = "both", amount = jitter,
    jitter = 0.2, offset = 1, density.par = list(na.rm = TRUE),
    horizontal = !vertical, vertical = TRUE, at, points = TRUE,
    pch = par("pch"), cex = par("cex"), col = "red", bg = "pink",
    lines = FALSE, line.lty = par("lty"), line.lwd = 0.5,
    line.col = "lightgray", add = FALSE, axes = TRUE,
    frame.plot = axes, axis.par, group.names, main = "", sub = "",
    xlab, ylab, dlab = "", glab = "", xlim, ylim, log = "")
```

Arguments

`formula`	a formula, such as `y ~ g`, where `y` is a numeric vector or matrix of data values to be split into groups according to the grouping variable `g` (usually a factor). A matrix is allowed for `y`. A formula of the form `y ~ 1` indicates no grouping.
`data`	a data frame (or list) from which the variables in `formula` should be taken.
`...`	For the formula method, further arguments to be passed to the default method. For the default method, additional data.
`subset`	an optional vector specifying a subset of observations to be used for plotting.
`na.action`	a function which indicates what should happen when the data contain NAs. The default is to ignore missing values in either the response or the group.
`x`	data, either a numeric vector, matrix or data frame (list).
`groups`	grouping variable (usually a factor).
`na.group`	logical, if false (the default) drop an 'NA' level from `groups`.
`method`	the method to be used to separate coincident points. The points are jitterd using kernel density estimates if `"violin"` (the default) is specified. The points are uniformly jittered if `"jitter"` is specified. The points are overplotted if `"overplot"` is specified. The coincident points are stacked if `"stack"` is specified. The last method only makes sense for very granular data.
`amount, jitter`	the amount of jitter when the method `"violin"` or `"jitter"` is used, the height of stack if the method `"stack"` is used. In the latter case, `NA` may be specified for the stack without a height adjustment. If `amount` is not specified, `jitter` can be used as the amount of jitter.
`offset`	when the method `"stack"` is used, points are stacked with the specified offset.
`density.par`	list of arguments passed to `density`.
`horizontal, vertical`	logical. If `horizontal` is `FALSE` (the default), the plots are drawn vertically with the first plot to the left. If `horizontal` is `TRUE`, the plots are drawn horizontally

with the first plot at the bottom. If horizontal is not specified, vertical can be used to set a graph orientation.

at — a numeric vector giving the locations where component plots should be drawn; defaults to 1:k where k is the number of components. Non-unique values can be specified. The argument at will be recycled to the number of components if necessary. See Details below.

side — the side on which the points are plotted, one of "both" (the default), "positive" or "negative". The argument side will be recycled to the number of components if necessary. See Details below.

points — logical, if true (the default) draw points.

pch, cex, col, bg — graphical parameters passed to points. Parameters will be recycled if necessary. See Details below.

lines — logical, if true draw lines in a similar way to parallel coordinates plots, defaults to FALSE.

line.lty, line.lwd, line.col — the line type, width and color for lines, passed to arrows. Parameters will be recycled if necessary. See Details below.

add — logical, if true *add* points or lines to current plot, defaults to FALSE.

axes — logical, whether to plot axes; defaults to TRUE.

frame.plot — logical, indicating if a frame (box) should be drawn; defaults to TRUE unless axes=FALSE is specified.

axis.par — list of arguments passed to axis. If specified, group.names is ignored.

group.names — labels for the group axis.

main, sub, xlab, ylab — character strings for annotation, passed to title.

dlab, glab — an alternative way to specify axis labels: see Details below.

xlim, ylim — numeric vectors of length 2, giving the range for the plot, passed to plot.window.

log — a character string, indicating if any axis should be drawn in log scale, passed to plot.window.

Details

The number of components k is the product of the number of (column) vectors n and the number of levels of group m. Arguments at and side are recycled to the length k. Graphical parameters pch, cex, col, bg, line.lty, line.lwd, line.col are recycled as follows. If the length of a parameter is greater than m, the parameter is recycled to the length k. Otherwise the parameter is recycled to the length m first, then each element of the parameter is repeated n times.

The dlab and glab labels may be used instead of xlab and ylab if those are not specified. dlab applies to the continuous data axis (the y axis if horizontal=FALSE or vertical=TRUE), glab to the group axis.

If add=TRUE, arguments axes, frame.plot, axis.par, group.names, main, sub, xlab, ylab, dlab, glab, xlim, ylim, log will be ignored.

Points are drawn after lines so that lines do not hide points.

Value

An invisible numeric vector, giving the coordinates of the center of each component on the group axis.

Author(s)

Yuki Atoh <yuki.atoh@gmail.com>

References

Hintze, J. L. and Nelson, R. D. (1998). *Violin plots: a box plot-density trace synergism.* The American Statistician, 52(2):181-184.

See Also

boxplot, stripchart, density, vioplot in package **vioplot**, beanplot in package **beanplot**, beeswarm in package **beeswarm**, parviol in package **parviol**

Examples

```
op <- par(mfrow=c(2,2))
gn <- c("SL", "SW", "PL", "PW")
for (i in c("overplot", "stack", "jitter", "violin"))
  viopoints(iris[1:4], method=i, horizontal=TRUE, col=2:5, group.names=gn,
    main=i)
par(op)

boxplot(Sepal.Length ~ Species, data=iris, ylab="Sepal.Length")
viopoints(Sepal.Length ~ Species, data=iris, col=2:4, add=TRUE)

boxplot(iris[1:4])
viopoints(iris[1:4], groups=iris[5], col=2:4, at=1:4+rep(c(-0.2,0,0.2),each=4),
  jitter=0.1, add=TRUE)
legend("topright", legend=levels(factor(iris[,5])), pch=1, col=2:4)

boxplot(iris[1:4])
viopoints(as.matrix(iris[1:4])~iris[,5], col=2:4, at=1:4,
  lines=TRUE, line.col=c("pink", "lightgreen", "skyblue"), add=TRUE)
legend("topright", legend=levels(factor(iris[,5])), pch=1, col=2:4)

viopoints(iris[c(1,3)], groups=iris[5], method="stack", amount=NA,
  at=rep(1:3,each=2), pch=rep(1:2,3), side=c("n","p"), col=rep(2:3,3),
  group.names=levels(factor(iris[,5])), main="stack")
legend("topleft", pch=1:2, col=2:3, legend=colnames(iris[c(1,3)]))


## Not run:
viopoints(as.matrix(iris[1:4]) ~ Species, data=iris, jitter=0.3, col=2:4,
  axis.par=list(at=1:12, labels=paste(rep(substr(levels(factor(iris[,5])),1,3),
  each=4), c("SL","SW","PL","PW"), sep="."), cex.axis=0.8, las=3))
## End(Not run)
```

Index