Package: beanplot (via r-universe)

November 2, 2024

Type Package

Title Visualization via Beanplots (Like Boxplot/Stripchart/Violin Plot)

Version 1.3.1

Date 2022-04-09

Author Peter Kampstra

Maintainer Peter Kampstra <beanplot@kampstra.net>

Description Plots univariate comparison graphs, an alternative to boxplot/stripchart/violin plot.

Suggests vioplot, lattice

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2022-04-09 07:22:29 UTC

Contents

---

beanplot-package   *Visualization via Beanplots (like Boxplot/Stripchart/Violin Plot*

Description

Plots univariate comparison graphs, alternative to boxplot/stripchart/violin plot

Details

Package:	beanplot
Type:	Package
Version:	1.3.1
Date:	2022-04-09
License:	GPL-2

The function beanplot does all the work.

Author(s)

Peter Kampstra <beanplot@kampstra.net>

References

Kampstra, P. (2008) Beanplot: A Boxplot Alternative for Visual Comparison of Distributions. *Journal of Statistical Software, Code Snippets*, **28**(1), 1-9. doi: 10.18637/jss.v028.c01

See Also

graphics vioplot

Examples

```
beanplot(rnorm(100), runif(100))
if (interactive()) vignette("beanplot", package = "beanplot")
```

---

beanplot	*Beanplot: A Boxplot/Stripchart/Vioplot Alternative*

Description

Plots beans to compare the distributions of different groups; it draws one bean per group of data. A bean consists of a one-dimensional scatter plot, its distribution as a density shape and an average line for the distribution. Next to that, an overall average for the whole plot is drawn per default.

Usage

```
beanplot(..., bw = "SJ-dpi", kernel = "gaussian", cut = 3, cutmin = -Inf,
         cutmax = Inf, grownage = 10, what = c(TRUE, TRUE, TRUE, TRUE),
         add = FALSE, col, axes = TRUE, log = "auto", handlelog = NA,
         ll = 0.16, wd = NA, maxwidth = 0.8, maxstripline = 0.96,
         method = "stack", names, overallline = "mean", beanlines = overallline,
         horizontal = FALSE, side = "no", jitter = NULL, beanlinewd = 2,
         frame.plot = axes, border = NULL, innerborder = NA, at = NULL,
         boxwex = 1, ylim = NULL, xlim = NULL, show.names = NA)
```

Arguments

...	data which to perform the beanplot on. This data can consist of dataframes, vectors and/or formulas. For each formula, a dataset can be specified with data=[dataset], and a subset can be specified with subset=[subset]. If subset/data arguments are passed, but there are not enough subset/data arguments, they are reused. Additionally, na.action, drop.unused.levels and xlev can be passed to `model.frame` in the same way. Also, parameters for `axis` and `title` can be passed.
bw	the bandwidth (method) being used, used by `density`. In case of a method, the average computed bandwidth is used.
kernel	see `density`.
cut	the beans are cut beyond cut*bw
cutmin	the low-ends of the beans are cut below mincut*bw. Defaults to cut.
cutmax	the high-ends of the beans are cut beyond maxcut*bw. Defaults to cut.
grownage	the width of a bean grows linearly with the count of points, until grownage is reached.
what	a vector of four booleans describing what to plot. In the following order, these booleans stand for the total average line, the beans, the bean average, and the beanlines. For example, what=c(0,0,0,1) produces a `stripchart`
add	if true, do not start a new plot
col	the colors to be used. A vector of up to four colors can be used. In the following order, these colors stand for the area of the beans (without the border, use `border` for that color), the lines inside the bean, the lines outside the bean, and the average line per bean. Transparent colors are supported. `col` can be a list of color vectors, the vectors are then used per bean, and reused if necessary.
axes	if false, no axes are drawn.
log	use log="y" or log="" to force a log-axis. In case of log="auto", a log-transformation is used if appropriate
handlelog	if handlelog then all the calculations are done using a log-scale. By default this is determined using the log parameter.
ll	the length of the beanline per point found.
wd	the linear transformation that determines the width of the beans. By default determined using maxwidth, and returned.
maxwidth	the maximum width of a bean.
maxstripline	the maximum length of a beanline.
method	the method used when two points on a bean are the same. "stack", "overplot" and "jitter" are supported.
names	a vector of names for the groups.
overallline	the method used for determining the overall line. Defaults to "mean", "median" is also supported.
beanlines	the method used for determining the average bean line(s). Defaults to "mean", "median" and "quantiles" are also supported.

horizontal	if true, the beanplot is horizontal
side	the side on which the beans are plot. Default is "no", for symmetric beans. "first", "second" and "both" are also supported.
jitter	passed to jitter as amount in case of method="jitter".
beanlinewd	the width used for the average bean line
frame.plot	if true, plots a frame.
border	the color for the border around a bean. NULL for par("fg"), NA for no border. If border is a vector, it specifies the color per bean, and colors are reused if necessary.
innerborder	a color (vector) for the border inside the bean(s). Especially useful if side="both". Use NA for no inner border. Colors are reused if necessary. The inner border is drawn as the last step in the drawing process.
at	the positions at which a bean should be drawn.
boxwex	a scale factor applied to all beans. Compatible with boxplot.
ylim	the range to plot.
xlim	the range to plot the beans at.
show.names	if true, plots the names as axis labels

Details

Use the "what" parameter to omit certain parts from drawing. Most parameters are compatible with boxplot and stripchart. For compatibility, arguments with the name "formula" or "x" are used as data. However, data or formulas do not need to be named "x" or "formula". The function handles (combinations of) dataframes, vectors and/or formulas.

Value

bw	the bandwidth (bw) used.
wd	the bean width (wd) used.
names	a vector of names for the groups.
stats	a vector of statistics calculated for the beanlines.
overall	statistic calculated for the overall line.
log	log axis that was selected.
ylim	ylim that was used.
xlim	xlim that was used.

Note

In case of more than 5000 values per bean, the autodetection of the log parameter is approximated. In such cases, use log="" or log="y". Also, the what parameter can help to omit parts that are not useful in certain plots, and take time to draw.

Author(s)

Peter Kampstra <beanplot@kampstra.net>

References

Kampstra, P. (2008) Beanplot: A Boxplot Alternative for Visual Comparison of Distributions. *Journal of Statistical Software, Code Snippets*, **28**(1), 1-9. doi: 10.18637/jss.v028.c01

See Also

boxplot, stripchart, density, rug, vioplot in package vioplot

Examples

```
beanplot(rnorm(22),rnorm(22),rnorm(22),main="Test!",rnorm(3))

#mostly examples taken from boxplot:
par(mfrow = c(1,2))
boxplot(count ~ spray, data = InsectSprays, col = "lightgray")
beanplot(count ~ spray, data = InsectSprays, col = "lightgray", border = "grey", cutmin = 0)

boxplot(count ~ spray, data = InsectSprays, col = "lightgray")
beanplot(count ~ spray, data = InsectSprays, col = "lightgray", border = "grey",
        overallline = "median")

boxplot(decrease ~ treatment, data = OrchardSprays,
        log = "y", col = "bisque", ylim = c(1,200))
beanplot(decrease ~ treatment, data = OrchardSprays,
        col = "bisque", ylim = c(1,200))

par(mfrow = c(2,1))
mat <- cbind(Uni05 = (1:100)/21, Norm = rnorm(100),
             T5 = rt(100, df = 5), Gam2 = rgamma(100, shape = 2))
par(las=1)# all axis labels horizontal
boxplot(data.frame(mat), main = "boxplot(*, horizontal = TRUE)",
        horizontal = TRUE, ylim = c(-5,8))
beanplot(data.frame(mat), main = "beanplot(*, horizontal = TRUE)",
        horizontal = TRUE, ylim = c(-5,8))

par(mfrow = c(1,2))
boxplot(len ~ dose, data = ToothGrowth,
        boxwex = 0.25, at = 1:3 - 0.2,
        subset = supp == "VC", col = "yellow",
        main = "Guinea Pigs' Tooth Growth",
        xlab = "Vitamin C dose mg",
        ylab = "tooth length", ylim = c(-1, 40), yaxs = "i")
boxplot(len ~ dose, data = ToothGrowth, add = TRUE,
        boxwex = 0.25, at = 1:3 + 0.2,
        subset = supp == "OJ", col = "orange")
legend("bottomright", bty="n", c("Ascorbic acid", "Orange juice"),
        fill = c("yellow", "orange"))
allplot <- beanplot(len ~ dose+supp, data = ToothGrowth,
    what=c(TRUE,FALSE,FALSE,FALSE),show.names=FALSE,ylim=c(-1,40), yaxs = "i")
beanplot(len ~ dose, data = ToothGrowth, add=TRUE,
        boxwex = 0.6, at = 1:3*2 - 0.9,
        subset = supp == "VC", col = "yellow",border="yellow2",
```

```
        main = "Guinea Pigs' Tooth Growth",
        xlab = "Vitamin C dose mg",
        ylab = "tooth length", ylim = c(3, 40), yaxs = "i",
        bw = allplot$bw, wd = allplot$wd, what = c(FALSE,TRUE,TRUE,TRUE))
beanplot(len ~ dose, data = ToothGrowth, add = TRUE,
        boxwex = 0.6, at = 1:3*2-0.1,
        subset = supp == "OJ", col = "orange",border="darkorange",
        bw = allplot$bw, wd = allplot$wd, what = c(FALSE,TRUE,TRUE,TRUE))
legend("bottomright", bty="n", c("Ascorbic acid", "Orange juice"),
        fill = c("yellow", "orange"))


par(mfrow = c(1,2))
boxplot(len ~ dose, data = ToothGrowth,
        boxwex = 0.25, at = 1:3 - 0.2,
        subset = supp == "VC", col = "yellow",
        main = "Guinea Pigs' Tooth Growth",
        xlab = "Vitamin C dose mg",
        ylab = "tooth length", ylim = c(-1, 40), yaxs = "i")
boxplot(len ~ dose, data = ToothGrowth, add = TRUE,
        boxwex = 0.25, at = 1:3 + 0.2,
        subset = supp == "OJ", col = "orange")
legend("bottomright", bty="n",c("Ascorbic acid", "Orange juice"),
        fill = c("yellow", "orange"))
beanplot(len ~ reorder(supp, len, mean) * dose, ToothGrowth,
        side = "b", col = list("yellow", "orange"), border = c("yellow2",
            "darkorange"), main = "Guinea Pigs' Tooth Growth",
        xlab = "Vitamin C dose mg", ylab = "tooth length", ylim = c(-1,
            40), yaxs = "i")
legend("bottomright", bty="n",c("Ascorbic acid", "Orange juice"),
        fill = c("yellow", "orange"))


#Example with multiple vectors and/or formulas
par(mfrow = c(2,1))
beanplot(list(all = ToothGrowth$len), len ~ supp, ToothGrowth, len ~ dose)
title("Tooth growth length (beanplot)")
#Trick using internal functions to do this with other functions:
mboxplot <- function(...){
    graphics::boxplot(beanplot:::getgroupsfromarguments(), ...)
}
mstripchart <- function(..., method = "overplot", jitter = 0.1, offset = 1/3,
            vertical = TRUE, group.names, add = FALSE,
            at = NULL, xlim = NULL, ylim = NULL,
            ylab = NULL, xlab=NULL, dlab = "", glab = "",
            log = "", pch = 0, col = par("fg"), cex = par("cex"),
            axes = TRUE, frame.plot = axes) {
    graphics::stripchart(beanplot:::getgroupsfromarguments(),
            method, jitter, offset, vertical, group.names, add,
            at, xlim, ylim, ylab, xlab, dlab, glab, log, pch, col, cex,
            axes, frame.plot)
}
mstripchart(list(all = ToothGrowth$len), len ~ supp, ToothGrowth, len ~ dose,
            xlim = c(0.5,6.5))
title("Tooth growth length (stripchart)")
```

Index