

# Package ‘plotpc’

September 27, 2015

**Version** 1.0.4

**Title** Plot Principal Component Histograms Around a Scatter Plot

**Author** Stephen Milborrow

**Maintainer** Stephen Milborrow <milbo@sonic.net>

**Depends** grid

**Description** Plot principal component histograms around a bivariate scatter plot.

**License** GPL-3

**URL** <http://www.milbo.users.sonic.net>

**Repository** CRAN

**Date/Publication** 2015-09-27 09:27:05

**NeedsCompilation** no

## R topics documented:

plotld	1
plotpc	2

<b>Index</b>	<b>10</b>
--------------	-----------

---

plotld	<i>Plot principal component loadings</i>
--------	--

---

### Description

Plot principal component loadings.

### Usage

```
plotld(x, npc=3, main="Loadings", lty=1, lwd=4 / 1:npc,  
col=gray(0:(npc-1) / npc), ylim=range(loadings), abs.=FALSE,  
cex=.8, ylab=if(abs.) "abs(loading)" else "loading",  
legend.x=NULL, legend.y=NULL)
```

**Arguments**

x	A matrix or dataframe, passed directly to <a href="#">princomp</a> .
npc	Number of principal components to plot. Default 3.
main	Plot title. Default "Loadings".
lty	Line type for for plotted lines. Default 1.
lwd	Line width of plotted lines. The default is ugly but effective: 4 / 1:npc.
col	Color of plotted lines. Default is a range of grays: gray(0:(npc-1) / npc).
ylim	Vertical limits of the graph. Default range(loadings).
abs.	Use absolute values of loadings. Default FALSE.
cex	Character expansion for axis and legend text. Default .8.
ylab	Default "loading".
legend.x, legend.y	Position of the legend. Default NULL, meaning automatic. For no legend, use an out-of-range legend.x or legend.y.

**See Also**

[princomp](#), [plotpc](#)

**Examples**

```
data(iris)
x <- iris[, -5] # -5 to drop Species
plotld(x)
```

---

plotpc

*Plot principal component histograms around a scatter plot*

---

**Description**

Plot principal component histograms around the scatter plot of two variables. Mostly useful as a tool for teaching principal components.

**Usage**

```
plotpc(x,
  xrange=NULL,
  hist=TRUE,
  main="Principal components",
  xlab=NULL,
  ylab=NULL,
  gp.points=gpar(cex=.6),
  pch=20,
  height=xrange/10,
```

```

breaks="Sturges",
adjust=1,
gp.hist=if(hist) gp.hist <- gpar(col="gray", fill="gray")
      else gp.hist <- gpar(col="black"),
gp.text=gpar(cex=.8, font=2),
gp.axis=gpar(col="gray", lwd=2),
sd.ellipse=NA,
gp.ellipse=gpar(col="gray", lwd=2),
heightx=NULL, breaksx=NULL, adjustx=NULL, gp.histx=NULL,
      textx="", gp.textx=NULL, axis.lenx=0, gp.axisx=NULL,
heighty=NULL, breaksy=NULL, adjusty=NULL, gp.histy=NULL,
      texty="", gp.texty=NULL, axis.leny=0, gp.axisy=NULL,
height1=NULL, flip1=FALSE,
      breaks1=NULL, adjust1=NULL, gp.hist1=NULL, offset1=NULL,
      text1=NULL, gp.text1=NULL, axis.len1=2, gp.axis1=NULL,
height2=NULL, flip2=FALSE,
      breaks2=NULL, adjust2=NULL, gp.hist2=NULL, offset2=NULL,
      text2=NULL, gp.text2=NULL, axis.len2=2, gp.axis2=NULL,
angle3=NA, height3=NULL, flip3=FALSE,
      breaks3=NULL, adjust3=NULL, gp.hist3=NULL, offset3=NULL,
      text3=NULL, gp.text3=NULL, axis.len3=0, gp.axis3=NULL,
angle4=NA, height4=NULL, flip4=FALSE,
      breaks4=NULL, adjust4=NULL, gp.hist4=NULL, offset4=NULL,
      text4=NULL, gp.text4=NULL, axis.len4=0, gp.axis4=NULL,
angle5=NA, height5=NULL, flip5=FALSE,
      breaks5=NULL, adjust5=NULL, gp.hist5=NULL, offset5=NULL,
      text5=NULL, gp.text5=NULL, axis.len5=0, gp.axis5=NULL,
angle6=NA, height6=NULL, flip6=FALSE,
      breaks6=NULL, adjust6=NULL, gp.hist6=NULL, offset6=NULL,
      text6=NULL, gp.text6=NULL, axis.len6=0, gp.axis6=NULL,
angle7=NA, height7=NULL, flip7=FALSE,
      breaks7=NULL, adjust7=NULL, gp.hist7=NULL, offset7=NULL,
      text7=NULL, gp.text7=NULL, axis.len7=0, gp.axis7=NULL,
yonx = FALSE, offset.yonx=-xrange/2.5,
      text.yonx="y~x", gp.text.yonx=NULL,
      axis.len.yonx=xrange/2.5, gp.axis.yonx=gpar(col=1),
xony = FALSE, offset.xony=-xrange/2.5,
      text.xony="x~y", gp.text.xony=NULL,
      axis.len.xony=xrange/2.5, gp.axis.xony=gpar(col=1))

```

## Arguments

Many users will find that they need only the first argument.

Use the `xrange` argument to add whitespace around the histograms.

Set `hist=FALSE` to plot densities rather than histograms.

Use `heightx` and the `height` arguments to adjust the height of histograms or to remove histograms from the plot.

Use `offset1` and the other `offset` arguments to adjust the positions of the histograms relative to the center of the graph.

Use `angle1` and the other angle arguments to add extra histograms to the plot at arbitrary angles.  
Use `yonx` and `xony` to add linear regression lines to the plot.

A two column matrix or dataframe. The principal components of the `x` will be calculated treating each column as a variable.

<code>hist</code>	Default TRUE to plot <a href="#">histograms</a> . Set to FALSE to plot <a href="#">densities</a> instead. The various "histogram" arguments will then apply to densities rather than to histograms.
<code>xrange</code>	The range of the x axis. That is, <code>xlim</code> will be <code>c(mean(x[,1]) - xrange/2, mean(x[,1]) + xrange/2)</code> , and <code>ylim</code> will have the same range about <code>mean(x[,2])</code> . Default NULL, meaning automatically deduce axis limits from the <code>x</code> argument.
<code>main</code>	Main title. Default "Principal components".
<code>xlab</code>	x axis label. Default NULL, meaning create the label automatically from the column names of <code>x</code> .
<code>ylab</code>	y axis label. Default NULL, meaning create the label automatically from the column names of <code>x</code> .
<code>gp.points</code>	Graphic parameters for the plotted points. Default <code>gpar(cex=.6)</code> .
<code>pch</code>	Plot character for the plotted points. Default 20.

**The following arguments apply to all histograms.** These can be overridden by using the histogram-specific argument e.g. override the height argument for the first principal component by specifying `height1`.

<code>height</code>	Height of histograms. Default <code>xrange/10</code> . Use a negative height to flip a histogram around its base.
<code>breaks</code>	Passed on to <a href="#">hist</a> . Default "Sturges". Using something like <code>breaks=12</code> can be useful.
<code>adjust</code>	Passed on to <a href="#">density</a> . Default 1. Use something like <code>adjust=.5</code> for more details in the density plots.
<code>gp.hist</code>	Graphic parameters for the histograms or densities. If <code>hist==TRUE</code> then the default is <code>gpar(col="gray", fill="gray")</code> where <code>col</code> is the color of the lines delineating the histograms, and <code>fill</code> is the color filling the histograms. If <code>hist==FALSE</code> then the default is <code>gpar(col="black")</code> .
<code>gp.axis</code>	Graphic parameters for the axis drawn through the scatter of points. Default <code>gpar(col="gray", lwd=2)</code> meaning draw the axes as thickish gray lines.
<code>sd.ellipse</code>	If greater than 0, draw a confidence ellipse for the principal components at <code>sd.ellipse</code> standard deviations. Default is NA, meaning do not draw an ellipse.
<code>gp.ellipse</code>	Graphic parameters for the ellipse. Default <code>gpar(col="gray", lwd=2)</code> .
<code>gp.text</code>	Graphic parameters for text above the histograms. Default <code>gpar(cex=.8, font=2)</code> .

**The following arguments apply to the histogram on the x axis.**

heightx	Default NULL, meaning use height. Use 0 to not plot the x histogram.
breaksx	Default NULL, meaning use breaks.
adjustx	Default NULL, meaning use adjust.
gp.histx	Default NULL, meaning use gp.hist.
textx	Text drawn above the histogram. Default "", meaning no text. The text is drawn using gp.textx.
gp.textx	Graphic parameters for the text above the histogram. Default NULL, meaning use gp.text.
axis.lenx	Length of horizontal line drawn through the center of the points. Units are standard deviations of $x[, 1]$ . Default 0, meaning do not plot a horizontal axis.
gp.axisx	Default NULL, meaning use gp.axis.

heighty, breaksy, adjusty, gp.histy, texty, gp.texty, axis.leny, gp.axisy  
As above but for the histogram on the y axis.

**The following arguments apply to the first principal component.**

height1	Default NULL, meaning use height. Use 0 to not plot the histogram for the first principal component.
flip1	Flip the position of the histogram around the axis of the first principal component. Default FALSE, meaning do not flip.
breaks1	Default NULL, meaning use breaks.
adjust1	Default NULL, meaning use adjust.
gp.hist1	Default NULL, meaning use gp.hist.
offset1	Distance of the histogram plot from the center of the graph, in native units. Default NULL, meaning automatic.
text1	Text drawn above the histogram. Default NULL, meaning generate the text automatically. Use "" for no text. The text is drawn using gp.text1.
gp.text1	Graphic parameters for the text above the histogram. Default NULL, meaning use gp.text.
axis.len1	Length of line drawn along the first principal axis. Units are standard deviations of the points projected onto that axis. Default 2, meaning draw a line of length plus and minus two standard deviations. Use 0 for no axis.
gp.axis1	Default NULL, meaning use gp.axis.

height2, flip2, breaks2, adjust2, gp.hist2, offset2, text2, gp.text2, axis.len2, gp.axis2  
As above but for the second principal component.

**The following arguments apply to the optional histogram at angle3.** By default, angle3=NA, meaning do not plot the histogram. Use, say, angle3=45 to plot a histogram at 45 degrees. By setting angle3 to angle7 you can plot up to five extra histograms at any angles.

angle3	Default NA, meaning do not plot a histogram. Use, say, angle3=45 to plot a histogram at 45 degrees.
height3	Default NULL, meaning use height.
flip3	Default FALSE.
breaks3	Default NULL, meaning use breaks.
adjust3	Default NULL, meaning use adjust.
gp.hist3	Default NULL, meaning use gp.hist.
offset3	Default NULL, meaning automatic.
text3	Default NULL, meaning automatic.
gp.text3	Default NULL, meaning use gp.text.
axis.len3	Length of axis drawn at angle3 through the scatter of points. Default 0, meaning do not plot the axis.
gp.axis3	Default NULL, meaning use gp.axis.

angle4, height4, flip4, breaks4, adjust4, gp.hist4, offset4, text4, gp.text4, axis.len4, gp.axis4  
As above but for the angle4 histogram.

angle5, height5, flip5, breaks5, adjust5, gp.hist5, offset5, text5, gp.text5, axis.len5, gp.axis5  
As above but for the angle5 histogram.

angle6, height6, flip6, breaks6, adjust6, gp.hist6, offset6, text6, gp.text6, axis.len6, gp.axis6  
As above but for the angle6 histogram.

angle7, height7, flip7, breaks7, adjust7, gp.hist7, offset7, text7, gp.text7, axis.len7, gp.axis7  
As above but for the angle7 histogram.

**The following arguments apply to the optional "y on x" regression line.**

yonx	TRUE to plot a "y on x" linear regression line. Default FALSE.
offset.yonx	Position of text plotted on regression line. Default $-xrange/2.5$ .
text.yonx	Text plotted on the regression line. Default "y~x".
gp.text.yonx	Graphic parameters for the text plotted on the regression line. Default NULL, meaning use gp.text.
axis.len.yonx	Length of regression line in <code>gpar</code> "native" units. Default $-xrange/2.5$ .
gp.axis.yonx	Graphic parameters for the regression line. Default <code>gpar(col=1)</code> .

xony, offset.xony, text.xony, gp.text.xony, axis.len.xony, gp.axis.xony  
As above but for a "x on y" regression.

**Value**

Invisibly returns the `viewport` used to create the plotpc axes. This allows you to add text using the "native" coordinates of the plot. See the examples below.

**Note**

Here is how to draw scatter plots for all pairs of principal components:

```
data(iris)
pc <- princomp(iris[, -5]) # -5 to drop Species
pairs(pc$scores, col=c(2,3,4)[unclass(iris$Species)])
```

**Author(s)**

Stephen Milborrow. Users are encouraged to send feedback — use milboATsonicPERIODnet  
<http://www.milbo.users.sonic.net>.

**See Also**

[plotld](#), [princomp](#), [hist](#), [density](#),

**Examples**

```

data(iris)
x <- iris[,c(3,4)] # select Petal.Length and Petal.Width
plotpc(x, main="Example 1\n")

# example with some parameters and showing densities
plotpc(x,
  main="Example 2:\nPrincipal component densities\n",
  hist=FALSE,           # plot densities not histograms
  adjust=.5,           # finer resolution in the density plots
  gp.axis=gpar(lty=3), # gpar of axes
  heightx=0,           # don't display x histogram
  heighty=0,           # don't display y histogram
  text1="Principal Component 1", # text above hist for 1st principal component
  text2="Principal Component 2", # text above hist for 2nd principal component
  axis.len2=4,         # length of 2nd principal axis (in std devs)
  offset1=2.5,         # offset of component 1 density plot
  offset2=5)           # offset of component 2 density plot

# example using "angles"
vp <- plotpc(x,
  main="Example 3:\nProjections\n",
  xrange=25,          # give ourselves some space
  heightx=0,          # don't display x histogram
  heighty=0,          # don't display y histogram
  angle3=-60,         # project at -60 degrees
  angle4=-25,         # project at -25 degrees
  angle5=20,          # project at 20 degrees
  angle6=70)          # project at 70 degrees

# add text to the graph, can use native coords
pushViewport(vp)
grid.text("Projections at\nvarious angles",
  x=unit(10, "native"), y=unit(12.5, "native"),
  gp=gpar(col="red"))
popViewport()

# example showing principal axes
x <- iris[iris$Species=="versicolor",c(3,4)]
vp <- plotpc(x,
  main="Example 4:\nPrincipal axes with confidence ellipse\n",
  sd.ellipse=2,       # ellipse at two standard devs
  heightx=0, heighty=0, height1=0, height2=0, # no histograms
  gp.ellipse=gpar(col=1), # ellipse in black
  axis.lenx=4, axis.leny=5, # lengthen horiz and vertical axes
  axis.len1=4, gp.axis1=gpar(col=1), # lengthen pc1 axis, draw in black
  axis.len2=8, gp.axis2=gpar(col=1)) # lengthen pc2 axis, draw in black

```



```
pushViewport(vp) # add text to the graph
un <- function(x) unit(x, "native")
grid.text("PC1", x=un(2.2), y=un(.6), gp=gpar(cex=.8, font=2))
grid.text("PC2", x=un(3.9), y=un(2.35), gp=gpar(cex=.8, font=2))
grid.text("X1", x=un(2.2), y=un(1.4), gp=gpar(cex=.8, font=2))
grid.text("X2", x=un(4.3), y=un(2.5), gp=gpar(cex=.8, font=2))
popViewport()

# example comparing linear regression to principal axis
x <- iris[iris$Species=="setosa",c(3,4)]
vp <- plotpc(x,
  main="Example 5:\nRegression lines and\nfirst principal component",
  heightx=0, heighty=0, height1=0, height2=0, # no histograms
  gp.points=gpar(col="steelblue"), # color of points
  axis.len1=4, gp.axis1=gpar(col="gray", lwd=3),
  axis.len2=.15, gp.axis2=gpar(col=1), # just a little blip of an axis
  yonx=TRUE, xony=TRUE) # display regression lines

pushViewport(vp) # add text to the principal component line
grid.text("PC1", x=unit(.8, "native"), y=unit(0, "native"),
  gp=gpar(col="gray", cex=.8, font=2))
popViewport()
```

# Index

densities, [4](#)  
density, [4](#), [8](#)  
  
gpar, [4](#), [7](#)  
  
hist, [4](#), [8](#)  
histograms, [4](#)  
  
plotld, [1](#), [8](#)  
plotpc, [2](#), [2](#)  
princomp, [2](#), [8](#)  
  
viewport, [7](#)