Package 'compactr'

February 19, 2015

Type Package

Title Creates empty plots with compact axis notation

Version 0.1

Date 2013-07-18

Author Carlisle Rainey

Maintainer Carlisle Rainey <carlislerainey@gmail.com>

Description Creates empty plots with compact axis notation to which users can add whatever they like (points, lines, text, etc.) The notation is more compact in the sense that the axis-labels and tick-labels are closer to the axis and the tick-marks are shorter. Also, if the plot appears as part of a matrix, the x-axis notation is suppressed unless the plot appears along the bottom row and the y-axis notation is suppress unless the plot appears along the left column.

License GPL-2

Collate 'eplot.R' 'addxaxis.R' 'compactr-package.R' 'mm.R' 'aplot.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2013-08-01 08:15:50

R topics documented:

compactr-package *Creates empty plots with compact axis notation*

Description

Creates empty plots with compact axis notation to which users can add whatever they like (points, lines, text, etc.) The notation is more compact in the sense that the axis-labels and tick-labels are closer to the axis and the tick-marks are shorter. Also, if the plot appears as part of a matrix, the x-axis notation is suppressed unless the plot appears along the bottom row and the y-axis notation is suppress unless the plot appears along the left column.

Details

Package:	compactr
Type:	Package
Version:	0.2
Date:	2013-07-18
License:	GPL-2

This package contains two functions. The most important, `eplot()`, creates an empty plot with compact axes that users can then draw on. The secondary function, `aplot()`, creates *another* plot using the same arguments as the previous call to `eplot()`, with the exception of main.

Author(s)

Maintainer: Carlisle Rainey <carlislerainey@gmail.com>

addxaxis *Add an x axis to the current plot*

Description

This function adds an x-axis to the current plot. Intended for use when the plot does not fall along the bottom row, but you plan to put no plot beneath it.

Usage

```
addxaxis()
```

Author(s)

Carlisle Rainey (e-mail, website)

Examples

```
par(mfrow = c(2,2), mar = c(.75,.75,.75,.75), oma = c(3,3,1,1))
eplot(xlim = c(-1, 1), ylim = c(-1, 1), xlab = "X Label")
aplot()
addxaxis()
aplot()
```

`aplot` *Create an empty plot with compact axis notation*

Description

The `aplot()` function simply calls `eplot()` again, using the same arguments (with the exception of `main`).

Usage

```
aplot(main = NULL)
```

Arguments

`main`	a label for the subplot. Intended for labeling a each plot in a matrix. If you need a title for the entire matrix of plots, or a single plot, I recommend using a call to the `mtext()` function.

Details

This function simply draws an empty plot with compact axis notation, to which the user can add points, lines, text, and so on. Also, if the plot appears as part of a matrix, the x-axis notation is suppressed unless the plot appears along the bottom row and the y-axis notation is suppress unless the plot appears along the left column.

Author(s)

Carlisle Rainey (e-mail, website)

Examples

```
# run these lines one at a time to see what happens
par(mfrow = c(2,2))
eplot(xlim = c(-1, 1), ylim = c(0, 10))
aplot(main = "Hey Look! No axis labels.")
aplot(main = "But this one has them?!")
aplot(main = "And this one does just what you'd expect!")
# after a call to eplot() or aplot(), I just add
# whatever I want to the plot.
```

eplot *Create an empty plot with compact axis notation*

Description

The eplot() function draws an empty plot to which the user can add points, lines, text, etc. The axis notation is more compact than the defaults for the plot() function. Also, axis and label are appropriately suppressed when the plot occurs as part of a matrix. The aplot() function simply calls eplot() again, using the same arguments (with the exception of main).

Usage

```
eplot(xlim, ylim, xlab = NULL, ylab = NULL, main = NULL,
  text.size = 1, tick.length = 0.02, xpos = -0.7,
  ypos = -0.5, xat = NULL, yat = NULL, xticklab = NULL,
  yticklab = NULL, xlabpos = 1.5, ylabpos = 1.5,
  annx = TRUE, anny = TRUE, box = TRUE)
```

Arguments

xlim	the x limits (x1, x2) of the plot.
ylim	the y limits of the plot.
xlab	a label for the x axis, defaults to empty space.
ylab	a label for the y axis, defaults to empty space.
main	a label for the subplot. Intended for labeling a each plot in a matrix. If you need a title for the entire matrix of plots, or a single plot, I recommend using a call to the mtext() function.
text.size	a numerical value giving the amount by which axis notation should be magnified. Reasonable values range from about 0.5 to 2.
tick.length	the length of tick marks as a fraction of the smaller of the width or height of the plotting region. Reasonable values range from about 0.01 to 0.1.
xpos,ypos	controls the distance from the tick labels to the axis. Reasonable values range from about -1 to 1.
xat,yat	the location of the tick marks along the axes. If "none," then the axis will not be annotated.
xticklab,yticklab	the labels for the tick marks. A character vector the length of xat and yat.
xlabpos,ylabpos	controls the distance from the axis labels to the axes. Reasonable values range from about 1 to 3.
annx,anny	include annotations for x and y axes?
box	should a box be plotted?

Details

This function simply draws an empty plot with compact axis notation, to which the user can add points, lines, text, and so on. Also, if the plot appears as part of a matrix, the x-axis notation is suppressed unless the plot appears along the bottom row and the y-axis notation is suppress unless the plot appears along the left column.

Author(s)

Carlisle Rainey (e-mail, website)

Examples

```
### Plot 0: illustrating the purpose

# run these lines one at a time to see what happens
par(mfrow = c(2,2))
eplot(xlim = c(-1, 1), ylim = c(0, 10))
aplot(main = "Hey Look! No axis labels.")
aplot(main = "But this one has them?!")
aplot(main = "And this one does just what you'd expect!")
# after a call to eplot() or aplot(), I just add
# whatever I want to the plot.


### Plot 1: a simple scatter plot

set.seed(1234)
x <- rnorm(100)
y <- x + rnorm(100)

par(mfrow = c(1,1), mar = c(3,3,1,1), oma = c(0,0,2,0))
eplot(xlim = c(min(x), max(x)), ylim = c(min(y), max(y)),
           xlab = "Explanatory Variable", ylab = "Outcome Variable")
points(x, y)
abline(lm(y ~ x), lwd = 3, col = "red")
mtext("A Clever Title", outer = TRUE)


### Plot 2: a matrix of scatter plots

# simulation multilevel data
set.seed(1234)
group <- rep(1:11, each = 15)
a <- rnorm(length(unique(group)), sd = 1)
b <- rnorm(length(unique(group)), mean = 1, sd = .3)
x <- rnorm(length(group))
y <- a[group] + x*b[group] + rnorm(length(group))

## estimate random effects models and pull out the estimates
#library(lme4)
#hier <- lmer(y ~ x + (1 + x | group))
#a.hat <- fixef(hier)[1] + ranef(hier)$group[, 1]
```

```
#b.hat <- fixef(hier)[2] + ranef(hier)$group[, 2]

# draw plot
par(mfrow = c(3,4), mar = c(.75,.75,.75,.75), oma = c(4,4,4,1))
for (i in 1:11) {
  eplot(xlim = c(min(x), max(x)), ylim = c(min(y), max(y)),
             xlab = "Explanatory Variable", ylab = "Outcome Variable",
             main = paste("Group", i))
  points(x[group == i], y[group == i])
  #abline(a = a.hat[i], b = b.hat[i])
  abline(lm(y[group == i] ~ x[group == i]), lty = 3)
}

# add an overall title
mtext("Comparing Partial Pooling and No Pooling", outer = TRUE, line = 2)


### Plot 3: a matrix of scatter plots using aplot() and addxaxis()

# use the same estimates as before

# draw the first plot with eplot()
par(mfrow = c(3,4), mar = c(.75,.75,.75,.75), oma = c(4,4,4,1))
eplot(xlim = c(min(x), max(x)), ylim = c(min(y), max(y)),
           xlab = "Explanatory Variable", ylab = "Outcome Variable",
           main = "Group 1")

# then add stuff
points(x[group == 1], y[group == 1])
#abline(a = a.hat[1], b = b.hat[1])
abline(lm(y[group == 1] ~ x[group == 1]), lty = 3)
legend(par("usr")[1], par("usr")[4],
       legend = c("partial pooling", "no pooling"), lty = c(1, 3),
       bty = "n", bg = NA, cex = .8)

# draw the rest with aplot()
for (i in 2:11) {
  aplot(main = paste("Group", i))
  # since we don't plan to have bottom right plot,
  # let's add an axis to the one above
  if (i == 9) { addxaxis() }
  points(x[group == i], y[group == i])
  #abline(a = a.hat[i], b = b.hat[i])
  abline(lm(y[group == i] ~ x[group == i]), lty = 3)
}
mtext("Comparing Partial Pooling and No Pooling", outer = TRUE, line = 2)
```

---

mm *Find the minimum and maximum of a vector*

Description

The function `mm()` finds the minimum and maximum of a vector. It is intended for use with `eplot()` to properly scale the axes.

Usage

```
mm(x)
```

Arguments

x	a vector

Author(s)

Carlisle Rainey (e-mail, website)

Examples

```
x <- rnorm(100)
y <- rnorm(100)

par(mfrow = c(1,1), mar = c(5,4,4,2), oma = c(0,0,0,0))
eplot(x, y, xlim = mm(x), ylim = mm(y))
```

Index