areaplot-package

Plot Stacked Areas and Confidence Bands as Filled Polygons

Description
Plot stacked areas and confidence bands as filled polygons, or add polygons to existing plots. A variety of input formats are supported, including vectors, matrices, data frames, formulas, etc.

Details
Plot:

\[
\begin{align*}
\texttt{areaplot} & \quad \text{stacked area} \\
\texttt{confplot} & \quad \text{confidence bands}
\end{align*}
\]

Author(s)
Arni Magnusson.

areaplot

Area Plot

Description
Produce a stacked area plot, or add polygons to an existing plot.

Usage
\[
\texttt{areaplot}(x, \ldots)
\]

## Default S3 method:
\[
\texttt{areaplot}(x, y = \text{NULL}, \text{prop} = \text{FALSE}, \text{rev} = \text{FALSE},
\quad \text{add} = \text{FALSE}, \text{xlab} = \text{NULL}, \text{ylab} = \text{NULL}, \text{col} = \text{NULL}, \text{legend} = \text{FALSE},
\quad \text{args.legend} = \text{NULL}, \ldots)
\]

## S3 method for class 'formula'
\[
\texttt{areaplot}(\text{formula}, \text{data}, \text{subset}, \text{na.action}, \text{xlab} = \text{NULL},
\quad \text{ylab} = \text{NULL}, \ldots)
\]

Arguments
\begin{itemize}
\item \texttt{x} \quad \text{a numeric vector of x values, or if y=\text{NULL} a numeric vector of y values. Can also be a 1-dimensional table (x values in names, y values in array), matrix or 2-dimensional table (x values in row names and y values in columns), a data frame (x values in first column and y values in subsequent columns), or a time-series object of class tsmts.}
\end{itemize}
Further arguments passed to `areaplot.default`, `matplot`, and `polygon`.

- **y**
  - A numeric vector of y values, or a matrix containing y values in columns.

- **prop**
  - Whether data should be plotted as proportions, so stacked areas equal 1.

- **rev**
  - Whether to plot the stacked areas from bottom to top, instead of top to bottom.

- **add**
  - Whether polygons should be added to an existing plot.

- **xlab**
  - A label for x axis.

- **ylab**
  - A label for y axis.

- **col**
  - Fill color of polygon(s). The default is a vector of gray colors.

- **legend**
  - A logical indicating whether a legend should be added, or a vector of strings for the legend. This only applies when more than one series is plotted.

- **args.legend**
  - A list of additional arguments to pass to the `legend` function.

- **formula**
  - A formula, such as `y~x`, `cbind(y1,y2)~x`, or `y~x+group`, specifying x and y values. A dot on the left-hand side, `~x`, means all variables except the one specified on the right-hand side.

- **data**
  - A data frame (or list) from which the variables in formula should be taken.

- **subset**
  - An optional vector specifying a subset of observations to be used.

- **na.action**
  - A function which indicates what should happen when the data contain NA values. The default is to ignore missing values in the given variables.

**Value**

Matrix of cumulative sums that was used for plotting.

**See Also**

- `polygon` is the underlying function used to draw polygons.
- `confplot` plots confidence bands as a filled area.
- `areaplot-package` gives an overview of the package.

**Examples**

```r
areaplot(rpois(10,40))
areaplot(rnorm(10))
# formula
areaplot(Armed.Forces~Year, data=longley)
areaplot(cbind(Armed.Forces,Unemployed)~Year, data=longley)
areaplot(.~Year, data=longley)
areaplot(circumference~age+Tree, Orange)
# add=TRUE
plot(1940:1970, 500*runif(31), ylim=c(0,500))
areaplot(Armed.Forces~Year, data=longley, add=TRUE)
# data frame
mydata <- longley[c("Year","GNP")]
```
areaplot(mydata)

# matrix
areaplot(WorldPhones)
areaplot(WorldPhones, prop=TRUE)

# table
require(MASS)
areaplot(table(Aids2$age))
areaplot(table(Aids2$age, Aids2$sex))

# ts/mts
areaplot(austres)
areaplot(Seatbelts[,c("drivers","front","rear")],
  ylab="Killed or seriously injured")
abline(v=1983+1/12, lty=3)

# legend
require(MASS)
areaplot(table(Aids2$age, Aids2$sex), legend=TRUE, col=c(2,4))
areaplot(table(Aids2$age, Aids2$sex), legend=TRUE, col=c(2,4), rev=TRUE)
wp <- WorldPhones[,order(colnames(WorldPhones))]
areaplot(wp, col=2:8, legend=TRUE, args.legend=list(x="topleft"))
areaplot(wp, col=2:8, legend=TRUE, args.legend=list(x="topleft"), rev=TRUE)

confplot

---

**confplot**  
*Plot Confidence Bands*

---

**Description**

Plot confidence bands of lower and upper y values as a filled area, or add polygon to an existing plot.

**Usage**

confplot(x, ...)

## Default S3 method:
confplot(x, y1 = NULL, y2 = NULL, add = FALSE,
  xlab = NULL, ylab = NULL, border = NA, col = "lightgray", ...)

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

**Arguments**

x a numeric vector of x values. Alternatively, x can be a matrix or data frame containing x values in the first column and lower and upper y values in the next two columns.
... further arguments passed to `confplot.default`, `matplot`, and `polygon`.

`y1` a numeric vector of lower y values. Alternatively, `y1` can be a matrix or data frame containing lower and upper y values in two columns.

`y2` a numeric vector of upper y values, if not already supplied in `x` or `y1`.

`add` whether confidence bands should be added to an existing plot.

`xlab` a label for x axis.

`ylab` a label for y axis.

`border` border color of polygon. The default `NA` is to omit borders.

`col` fill color of polygon.

`formula` a formula, such as `cbind(y1,y2)`~`x`, specifying x and y values.

`data` a data frame (or list) from which the variables in formula should be taken.

`subset` an optional vector specifying a subset of observations to be used.

`na.action` a function which indicates what should happen when the data contain `NA` values. The default is to ignore missing values in the given variables.

Value
Data frame of coordinates that were used for plotting.

See Also

`polygon` is the underlying function used to draw polygons.

`areaplot` produces a stacked area plot.

`areaplot-package` gives an overview of the package.

The `gplots` and `plotrix` packages provide functions to plot error bars.

Examples

```r
model <- lm(log(dist)~log(speed), cars)
ci95 <- predict(model, data.frame(speed=4:25), interval="confidence")
ci50 <- predict(model, data.frame(speed=4:25), interval="confidence", level=0.5)
x <- log(4:25)
y1 <- ci95[ , "lwr"]
y2 <- ci95[ , "upr"]
mydata <- data.frame(x, y1, y2)

# Input format
confplot(x, y1, y2) # vectors
confplot(x, cbind(y1,y2)) # y values in 2 columns
confplot(mydata) # data in 3 columns
confplot(cbind(y1,y2)~x, mydata) # formula

# Overlay
plot(log(dist)~log(speed), cars, type="n")
confplot(x, ci95[,2:3], add=TRUE)
confplot(x, ci50[,2:3], add=TRUE, col="darkgray")
```
lines(x, ci95[,1])
points(log(dist)-log(speed), cars)
Index

areaplot, 2, 2, 5
areaplot-package, 2
confplot, 2, 3, 4
formula, 3, 5
legend, 3
polygon, 3, 5