

Package ‘FishMaps’

July 28, 2012

Title Proportional symbol mapping for fishery data

Version 0.2-0

Date 2007-03-14

Author Fernando Mayer

Maintainer Fernando Mayer <fernandomayer@gmail.com>

Depends R (>= 2.4.1), maps, mapdata, sp

Description Proportional symbol mapping for fishery data (catch, effort and CPUE), using pre-defined or user-defined maps.

License GPL-3

R topics documented:

BB.data.y	2
BB.data.yq	2
LL.data.y	3
LL.data.yq	4
mapBB	4
mapLL	6
mapq	7
mapy	10
Index	13

BB.data.y	<i>Baitboat yearly aggregated data</i>
-----------	--

Description

Skipjack tuna CPUE by year, caught by the brazilian baitboat fleet, based at Itajai (SC) harbor.

Usage

```
data(BB.data.y)
```

Format

A data frame with 56 observations on the following 4 variables.

year a factor with levels 2001 2002

lat a numeric vector

lon a numeric vector

cpue a numeric vector

Source

Randomly generated data.

Examples

```
data(BB.data.y)
str(BB.data.y)
```

BB.data.yq	<i>Baitboat quarterly aggregated data</i>
------------	---

Description

Skipjack tuna CPUE by quarter and year, caught by the brazilian baitboat fleet, based at Itajai (SC) harbor.

Usage

```
data(BB.data.yq)
```

Format

A data frame with 120 observations on the following 5 variables.

year a factor with levels 2001 2002

quarter a factor with levels 1 2 3 4

lat a numeric vector

lon a numeric vector

cpue a numeric vector

Source

Randomly generated data.

Examples

```
data(BB.data.yq)
str(BB.data.yq)
```

LL.data.y

Longline yearly aggregated data

Description

Swordfish CPUE by year, caught by the brazilian longline fleet, based at Itajai (SC) harbor.

Usage

```
data(LL.data.y)
```

Format

A data frame with 82 observations on the following 4 variables.

year a factor with levels 2001 2002 2003 2004 2005

lat a numeric vector

lon a numeric vector

cpue a numeric vector

Source

Randomly generated data.

Examples

```
data(LL.data.y)
str(LL.data.y)
```

LL.data.yq

Longline quarterly aggregated data

Description

Swordfish CPUE by year and quarter, caught by the brazilian longline fleet, based at Itajai (SC) harbor.

Usage

```
data(LL.data.yq)
```

Format

A data frame with 181 observations on the following 5 variables.

year a factor with levels 2001 2002 2003 2004 2005

quarter a factor with levels 1 2 3 4

lat a numeric vector

lon a numeric vector

cpue a numeric vector

Source

Randomly generated data.

Examples

```
data(LL.data.yq)
str(LL.data.yq)
```

mapBB

Draw a map for baitboat fishery data

Description

Draw a standard map where baitboat fishery data (catch, effort and CPUE) from the south brazilian coast can be plotted. Maps are built with the standard function map from the **maps** package. This function is used by other functions, such as [mapy](#) and [mapq](#).

Usage

```
mapBB(grid = TRUE, names = FALSE, map.data = c("world", "worldHires"), ...)
```

Arguments

grid	logical. If TRUE (the default), a grid is drawn in 5 x 5 squares.
names	logical. if TRUE, the name of coastal countries (i.e. Brazil) are drawn in upper case.
map.data	the database used to draw the maps. This should be one of “world” (default - low resolution) or “worldHires” (hi resolution). See more details in ?map.
...	other arguments from the map function. See ?map.

Details

The function mapBB only draw a pre-defined map of the south brazilian coast. The boundaries for this map are: from 20 S to 35 S of latitude and from 40 W to 55 W of longitude.

Value

A map from the south brazilian coast.

Author(s)

Fernando Mayer <fernandomayer@gmail.com>

References

Becker, R. A. and Wilks, A. R. 1993. Maps in S. *AT&T Bell Laboratories Statistics Research Report [93.2]*.

See Also

map from package **maps**. degAxis from **sp** package, from which the map axes were drawn with the degree symbol.

Examples

```
# a simple example
mapBB(grid = FALSE, names = TRUE)
# mapBB with full resolution
mapBB(map.data = "worldHires", resolution = 0)
```

mapLL

*Draw a map for longline fishery data***Description**

Draw a standard map where longline fishery data (catch, effort and CPUE) from the southwestern Atlantic Ocean can be plotted. Maps are built with the standard function `map` from the **maps** package. This function is used by other functions, such as `mapy` and `mapq`.

Usage

```
mapLL(grid = TRUE, names = FALSE, map.data = c("world", "worldHires"), ...)
```

Arguments

<code>grid</code>	logical. If TRUE (the default), a grid is drawn in 5 x 5 squares.
<code>names</code>	logical. if TRUE, the name of coastal countries (i.e. Brazil, Uruguay and Argentina) are drawn in upper case. Uruguay and Argentina are abbreviated.
<code>map.data</code>	the database used to draw the maps. This should be one of “world” (default - low resolution) or “worldHires” (hi resolution). See more details in <code>?map</code> .
<code>...</code>	other arguments from the <code>map</code> function. See <code>?map</code> .

Details

The function `mapLL` only draw a pre-defined map of the southwestern Atlantic Ocean. Nothing more is done. The boundaries for this map are: from 10 S to 55 S of latitude and from 20 W to 65 W of longitude.

Value

A map from the southwestern Atlantic Ocean.

Author(s)

Fernando Mayer <fernandomayer@gmail.com>

References

Becker, R. A. and Wilks, A. R. 1993. Maps in S. *AT&T Bell Laboratories Statistics Research Report [93.2]*.

See Also

`map` from package **maps**. `degAxis` from **sp** package, from which the map axes were drawn with the degree symbol.

Examples

```
# a simple example
mapLL(grid = FALSE, names = TRUE)
# mapLL with full resolution
mapLL(map.data = "worldHires", resolution = 0)
```

mapq

Plots quarterly fishery data into maps

Description

Plots georeferenced fishery data (catch, effort and CPUE) aggregated by quarter and year into maps. This function can use pre-defined maps, such as [mapLL](#) and [mapBB](#), or users can specify their own map boundaries. Data are properly separated to form classes and each class has a weight. Solid circles proportional to the class weights are plotted in the map. A legend is also provided. Resulting maps can be displayed on the screen (each one after the user presses the <Enter> key), or automatically saved in PNG or PDF file format, with an appropriate name.

Usage

```
mapq(year, quarter, lat, lon, cfu, breaks, type = c("LL", "BB"),
      ident = TRUE, ident.type = c("num", "let"), ident.cex = 1,
      xlim = NULL, ylim = NULL, majortick = NULL, minortick = NULL,
      mapgrid = TRUE, legend = TRUE, leg.pos = "bottomright",
      leg.cex = 1, leg.title = NULL, fig = FALSE,
      fig.type = c("png", "pdf"), fig.w, fig.h, fig.name,
      fig.par = NULL, ...)
```

Arguments

year	a vector with years relative to data. This can be numeric or factor.
quarter	a vector with quarters relative to data. This can be numeric or factor.
lat	a numeric vector with latitudes in degrees. Southern positions from Equator must be negative.
lon	a numeric vector with longitudes in degrees. Western positions from Greenwich must be negative.
cfu	a numeric vector with catch (c), effort (f), or CPUE (u) data.
breaks	either a numeric vector of cut points or number giving the number of intervals which cfu is to be cut into. This is the same used in the function <code>cut</code> . See <code>?cut</code> for more details. NOTE: if cfu contains zero values, the best way to specify breaks is only with the number of intervals. If you wish to specify a numeric vector of cut points, start with a number different of zero. This function was designed to handle zero values in cfu properly.

type	the type of map to be drawn. “LL” (the default) uses the mapLL function, and “BB” uses the mapBB function. Only specify one of them if you want to use the pre-defined maps. If you will specify your own map boundaries, don’t specify the type and use the xlim and ylim arguments (see below).
ident	logical. If TRUE (the default), add an identifier for the map.
ident.type	the type of identifier to be used. “num” uses the very same years as in the year vector. “let” will add an uppercase letter, starting with A and so on. The identifier will always be located in the upper left corner of the map.
ident.cex	a numeric value with the character expansion of the identifier.
xlim	a numeric vector of length 2, giving the range of longitudes in which the map will be restricted. Use xlim only if you want to specify your own map boundaries.
ylim	a numeric vector of length 2, giving the range of latitudes in which the map will be restricted. Use ylim only if you want to specify your own map boundaries.
majortick	a number specifying the interval at which major ticks will be drawn in the map. This will also be the interval at which the number of latitude and longitude will be explicitly displayed.
minortick	a number specifying the interval at which minor ticks will be drawn in the map. This will also be the interval at which the number of latitude and longitude will NOT be explicitly displayed.
mapgrid	should a grid be drawn in the user-defined map? Note that mapgrid is only used when xlim and ylim are specified, i.e, it’s a user-defined map. To control the grid output in a pre-defined map (when using argument type), use the map.grid argument which is used in the mapLL and mapBB functions.
legend	logical. Should a legend be displayed in the map?
leg.pos	the legend position. A character name indicating where the legend should be placed in the map. Must be one of “bottomright”, “bottom”, “bottomleft”, “left”, “topleft”, “top”, “topright”, “right”, “center”.
leg.cex	the character expansion for the legend. This is intended to be used in the legend title and in the legend itself.
leg.title	a character title for the legend.
fig	logical. Should mapq make figures of the maps instead of plotting them on the screen? Note that if this is set to TRUE, you will not see the maps on the screen.
fig.type	the file format for the figures. Must be one of “png” (default) or “pdf”.
fig.w	the figure width. For “png” use pixels, and for “pdf” use inches.
fig.h	the figure height. For “png” use pixels, and for “pdf” use inches.
fig.name	a character name. This name is used to compose the name of the file generated by mapy. The name of the files are composed of map_fig.name_year.ext. So if fig.type is “png”, and fig.name is, e.g., “cpue”, a typical file name will be map_cpue_2001.png. The ‘year’ will vary with the years as in year. Use fig.name to distinguish your variables in the file names.
fig.par	a list specifying the graphical parameters for the figure. Any arguments used in par() are allowed. See ?par for more details.
...	other arguments from the map and mapLL (or mapBB) functions. See ?map, ?mapLL and ?mapBB.

Details

Coming soon.

Value

This function plots maps with solid circles proportional to a variable (catch, effort or CPUE) on the current graphics device. Each map is shown after the <Enter> key is pressed by the user. If `fig = TRUE` then the maps are not seen on the screen, but are saved to files (as PNG or PDF file formats) in the working directory.

Author(s)

Fernando Mayer <fernandomayer@gmail.com>

References

Becker, R. A. and Wilks, A. R. 1993. Maps in S. *AT&T Bell Laboratories Statistics Research Report [93.2]*.

See Also

map from the **maps** package. [mapLL](#) and [mapBB](#) from **FishMaps**. `cut`, which is used to separate data into classes.

Examples

```
## loading longline data
data(LL.data.yq)
attach(LL.data.yq)
# a simple example
mapq(year,quarter,lat,lon,cpue,breaks=4)
# an enhanced map: full resolution, legend title and no grid
mapq(year,quarter,lat,lon,cpue,breaks=4,map.data="worldHires",
      resolution=0,grid=FALSE,leg.title="CPUE\n(kg/1000 hooks)")
# a user-defined map
mapq(year,quarter,lat,lon,cpue,breaks=4,xlim=c(-70,-15),ylim=c(-60,-5),
      majortick=10,minortick=5,border=0)
detach(LL.data.yq)

## loading baitboat data
data(BB.data.yq)
attach(BB.data.yq)
# a simple example
mapq(year,quarter,lat,lon,cpue,type="BB",breaks=4)
# an enhanced map: full resolution, legend title and no grid
mapq(year,quarter,lat,lon,cpue,type="BB",breaks=4,map.data="worldHires",
      resolution=0,grid=FALSE,leg.title="CPUE\n(kg/fishing day)")
# a user-defined map
mapq(year,quarter,lat,lon,cpue,breaks=4,xlim=c(-52,-40),ylim=c(-34,-22),
      majortick=2,minortick=1,border=0)
```

```
detach(BB.data.yq)
```

mapy

Plots yearly fishery data into maps

Description

Plots georeferenced fishery data (catch, effort and CPUE) aggregated by year into maps. This function can use pre-defined maps, such as [mapLL](#) and [mapBB](#), or users can specify their own map boundaries. Data are properly separated to form classes and each class has a weight. Solid circles proportional to the class weights are plotted in the map. A legend is also provided. Resulting maps can be displayed on the screen (each one after the user presses the <Enter> key), or automatically saved in PNG or PDF file format, with an appropriate name.

Usage

```
mapy(year, lat, lon, cfu, breaks, type = c("LL", "BB"),
      ident = TRUE, ident.type = c("num", "let"),
      ident.cex = 1, xlim = NULL, ylim = NULL,
      majortick = NULL, minortick = NULL, mapgrid = TRUE,
      legend = TRUE, leg.pos = "bottomright", leg.cex = 1,
      leg.title = NULL, fig = FALSE, fig.type = c("png", "pdf"),
      fig.w, fig.h, fig.name, fig.par = NULL, ...)
```

Arguments

year	a vector with years relative to data. This can be numeric or factor.
lat	a numeric vector with latitudes in degrees. Southern positions from Equator must be negative.
lon	a numeric vector with longitudes in degrees. Western positions from Greenwich must be negative.
cfu	a numeric vector with catch (c), effort (f), or CPUE (u) data.
breaks	either a numeric vector of cut points or number giving the number of intervals which cfu is to be cut into. This is the same used in the function cut. See ?cut for more details. NOTE: if cfu contains zero values, the best way to specify breaks is only with the number of intervals. If you wish to specify a numeric vector of cut points, start with a number different of zero. This function was designed to handle zero values in cfu properly.
type	the type of map to be drawn. "LL" (the default) uses the mapLL function, and "BB" uses the mapBB function. Only specify one of them if you want to use the pre-defined maps. If you will specify your own map boundaries, don't specify the type and use the xlim and ylim arguments (see below).
ident	logical. If TRUE (the default), add an identifier for the map.

<code>ident.type</code>	the type of identifier to be used. “num” uses the very same years as in the year vector. “let” will add an uppercase letter, starting with A and so on. The identifier will always be located in the upper left corner of the map.
<code>ident.cex</code>	a numeric value with the character expansion of the identifier.
<code>xlim</code>	a numeric vector of length 2, giving the range of longitudes in which the map will be restricted. Use <code>xlim</code> only if you want to specify your own map boundaries.
<code>ylim</code>	a numeric vector of length 2, giving the range of latitudes in which the map will be restricted. Use <code>ylim</code> only if you want to specify your own map boundaries.
<code>majortick</code>	a number specifying the interval at which major ticks will be drawn in the map. This will also be the interval at which the number of latitude and longitude will be explicitly displayed.
<code>minortick</code>	a number specifying the interval at which minor ticks will be drawn in the map. This will also be the interval at which the number of latitude and longitude will NOT be explicitly displayed.
<code>mapgrid</code>	should a grid be drawn in the user-defined map? Note that <code>mapgrid</code> is only used when <code>xlim</code> and <code>ylim</code> are specified, i.e, it’s a user-defined map. To control the grid output in a pre-defined map (when using argument <code>type</code>), use the <code>map.grid</code> argument which is used in the mapLL and mapBB functions.
<code>legend</code>	logical. Should a legend be displayed in the map?
<code>leg.pos</code>	the legend position. A character name indicating where the legend should be placed in the map. Must be one of “bottomright”, “bottom”, “bottomleft”, “left”, “topleft”, “top”, “topright”, “right”, “center”.
<code>leg.cex</code>	the character expansion for the legend. This is intended to be used in the legend title and in the legend itself.
<code>leg.title</code>	a character title for the legend.
<code>fig</code>	should mapy make figures of the maps instead of plotting them on the screen? Note that if this is set to TRUE, you will not see the maps on the screen.
<code>fig.type</code>	the file format for the figures. Must be one of “png” (default) or “pdf”.
<code>fig.w</code>	the figure width. For “png” use pixels, and for “pdf” use inches.
<code>fig.h</code>	the figure height. For “png” use pixels, and for “pdf” use inches.
<code>fig.name</code>	a character name. This name is used to compose the name of the file generated by mapy. The name of the files are composed of <code>map_fig.name_year.ext</code> . So if <code>fig.type</code> is “png”, and <code>fig.name</code> is, e.g., “cpue”, a typical file name will be <code>map_cpue_2001.png</code> . The ‘year’ will vary with the years as in <code>year</code> . Use <code>fig.name</code> to distinguish your variables in the file names.
<code>fig.par</code>	a list specifying the graphical parameters for the figure. Any arguments used in <code>par()</code> are allowed. See <code>?par</code> for more details.
<code>...</code>	other arguments from the <code>map</code> and mapLL (or mapBB) functions. See <code>?map</code> , <code>?mapLL</code> and <code>?mapBB</code> .

Details

Coming soon.

Value

This function plots maps with solid circles proportional to a variable (catch, effort or CPUE) on the current graphics device. Each map is shown after the <Enter> key is pressed by the user. If `fig = TRUE` then the maps are not seen on the screen, but are saved to files (as PNG or PDF file formats) in the working directory.

Author(s)

Fernando Mayer <fernandomayer@gmail.com>

References

Becker, R. A. and Wilks, A. R. 1993. Maps in S. *AT&T Bell Laboratories Statistics Research Report [93.2]*.

See Also

map from the **maps** package. [mapLL](#) and [mapBB](#) from **FishMaps**. `cut`, which is used to separate data into classes.

Examples

```
## loading longline data
data(LL.data.y)
attach(LL.data.y)
# a simple example
mapy(year,lat,lon,cpue,breaks=4)
# an enhanced map: full resolution, legend title and no grid
mapy(year,lat,lon,cpue,breaks=4,map.data="worldHires",resolution=0,
      grid=FALSE,leg.title="CPUE\n(kg/1000 hooks)")
# a user-defined map
mapy(year,lat,lon,cpue,breaks=4,xlim=c(-70,-15),ylim=c(-60,-5),
      majortick=10,minortick=5,border=0)
detach(LL.data.y)

## loading baitboat data
data(BB.data.y)
attach(BB.data.y)
# a simple example
mapy(year,lat,lon,cpue,type="BB",breaks=4)
# an enhanced map: full resolution, legend title and no grid
mapy(year,lat,lon,cpue,type="BB",breaks=4,map.data="worldHires",
      resolution=0,grid=FALSE,leg.title="CPUE\n(t/fishing day)")
# a user-defined map
mapy(year,lat,lon,cpue,breaks=4,xlim=c(-52,-40),ylim=c(-34,-22),
      majortick=2,minortick=1,border=0)
detach(BB.data.y)
```

Index

*Topic **datasets**

BB.data.y, [2](#)

BB.data.yq, [2](#)

LL.data.y, [3](#)

LL.data.yq, [4](#)

*Topic **hplot**

mapBB, [4](#)

mapLL, [6](#)

*Topic **iplot**

mapq, [7](#)

mapy, [10](#)

BB.data.y, [2](#)

BB.data.yq, [2](#)

LL.data.y, [3](#)

LL.data.yq, [4](#)

mapBB, [4](#), [7–12](#)

mapLL, [6](#), [7–12](#)

mapq, [4](#), [6](#), [7](#)

mapy, [4](#), [6](#), [10](#)