Package ‘blighty’

February 19, 2015

Version 3.1-4
Date 2012/04/18
Title United Kingdom coastlines
Author David Lucy <d.lucy@lancaster.ac.uk>
Maintainer David Lucy <d.lucy@lancaster.ac.uk>
Description Function for drawing the coastline of the British Isles
Depends utils
License GPL-2
Repository CRAN
Date/Publication 2012-04-17 14:45:23
NeedsCompilation no

R topics documented:

  blighty ................................................................. 1
  map.scale ............................................................ 5
  north.pointer ......................................................... 6

Index

blighty British Isles Coastlines

Description

  Function for drawing the coastlines of the British Isles

Usage

  blighty(place="set.British.Isles", set=TRUE, grid=FALSE, xlimits, ylimits, xpadding=0, ypadding=0, pa...
Arguments

place  map set - a list of objects to plot - see notes - default - "set.British.Isles"
set    set to FALSE if place is not a standard set of objects - default - TRUE
grid   TRUE or FALSE - places a grid in OS km over the map - default - FALSE
xlims  vector of limits in OS km in which to draw the map
ylims  vector of limits in OS km in which to draw the map
xpadding  kms to leave to either side between the edges and map in the x direction - default = 0
ypadding  kms to leave to either side between the edges and map in the y direction - default = 0
parcol  primary area colour - "transparent" for no shading - default - par("bg")
parbor  primary area colour - "transparent" for invisible - default - par("fg")
parwdh  primary area border width - default 1
sarcol  secondary area colour - "transparent" for no shading - default - par("bg")
sarbor  secondary area colour - "transparent" for invisible - default - par("fg")
sarwdh  secondary area border width - default 1
parang  angle of hatching lines for primary areas - default NULL - parden must be set to positive if hatching is to appear - look at polygon for further information
parden  density of hatching lines for primary areas in lines per inch - default - NULL - parden must be set to positive if hatching is to appear - look at polygon for further information
sarang  angle of hatching lines for secondary areas - default NULL - parden must be set to positive if hatching is to appear - look at polygon for further information
sarden  density of hatching lines for secondary areas in lines per inch - default - NULL - parden must be set to positive if hatching is to appear - look at polygon for further information
tlncol  colour of lines for non-area objects - default - par("fg")
tlnwdh  width of lines for non-area objects - default = 1
grdcol  colour of grid lines - default - par("fg")
grdwdh  width of grid lines - default = 1

Value

returns the object blighty.mapinfo in the global environment to give information for north.pointer and map.scale

Acknowledgements

Dr. Breedette Hayes of Dublin, for the Irish coordinates, and Dr. Mandy Jay of the Department of Archaeology, University of Sheffield, for some of the detail work on the Scottish Western Isles.
Note

Inspired by the package oz, blighty() draws the coastline of many of the British Isles. The output is suitable for general purpose illustrations, distribution diagrams; it is not heavyweight GIS or meant for calculation in any way.

The original coordinates were taken from a scan of a map of England, Scotland and Wales in my 1936 edition of The World of Wonder to avoid any copyright problems. The scan was then imported into xfig and the coastline and internal boundaries were digitised using the polyline function. Each section of coastline was then saved as separate file. A small amount of hand editing of each file allows them to be read in as coordinate points in R (functions available from the author and in the /misc directory of the package). The problem with using such an early work from which to take the primary image is that bits which were present then may not be now as coastal erosion may have affected the East of England.

The coordinate system was taken from various easily recognisable points of the UK coastline. The calculated error on these is in 100s of metres, the real error from comparison is about 3km. Still, most features for which a grid reference from an Ordnance Survey map can be plotted on easily.

For blighty_3-0.0 Irish features have been added. However, the Irish Ordnance Survey grid and the United Kingdom Ordnance Survey grid are entirely different, so were difficult to tie together. See the text file in the /misc directory. blighty_3-1.0 has some of the Scottish Isles revised.

Several people had pointed out that Jura was missing.

Don’t use the xlims and ylims to zoom in too far as the point resolution isn’t really up to it. There is now a warning for this - basically the maps start to look bad at anything less than 200km.

It can also take a few seconds when drawing all of the British Isles, this is mainly because of the west coast of Scotland and Western Isles which because of their shape take up loads of data points, the West coast of Ireland only adds to the delay.

At the request of numerous people there are now a set of primitive tools in the package /misc directory to enable people to construct and use their own maps. To get this download the source blighty_x.y-z.tar.gz to a directory, tar -zxvf it, then look in the blighty_x.y-z.tar.gz/misc directory - although it is part of the distribution it is not installed as such, therefore needs manual unpacking.

Basically blighty is an elaborate front end for polygon and points with some data files added. As such it will plot more or less any sets of coordinate points - so is by no means restricted to the distributed objects. At the moment it has three object types (although more can easily be added), one is a primary area - used for a landmass, the secondary area is a shape within the landmass, such as a lake, the tertiary feature type is a non-area type such as a river. The type of feature is encoded into the data files, blighty() then knows how to treat the object. Unfortunately secondary area objects will not cut out an area of the primary objects - so unless the parent primary area object is "transparent", the secondary area object can be "transparent", but it will merely show the primary area colour. It would be good to rectify this, but I can see no way at the moment of doing so with the present data structure.

Additional place sets are: "set.UK", "set.England", "set.Wales", set.England.Wales,"set.Scotland", "set.British.Isles", "set.Ireland" and "set.Scotland.all", although it is simple to specify your own - see additional documentation in /misc directory, or to select new combinations of existing features from the feature sets listed above. To do this you simply have to specify a vector of strings which refer to the individual objects and send that as the place parameter. You can easily pull these out from the standard sets which are now loaded at invocation time.
Author(s)

David Lucy <<d.lucy@lancaster.ac.uk>> http://www.maths.lancs.ac.uk/~lucy/

See Also

map.scale north.pointer oz

Examples

blightly() # British Isles coastline
blightly(grid=TRUE) # British Isles coastline with grid
points(399,91) # add in Poole OS coords last digit is 1km
text(400,50,"Poole")

blightly(place="set.Scotland") # Scotland and the Western Isles
points(327, 672) # OS coords for Authors Seat
text(327, 655, "Edinburgh")

blightly(place="set.Wales") # Wales and Anglesey
blightly(place="set.Ireland") # Ireland

blightly(place="set.England", xlims=c(400,600), ylims=c(50, 200))
box() # south of England and the Isle of Wight
# and the Thames Estuary

blightly(place="set.British.Isles") # do a plot then add the capital cities
x <- c(532, 327, 317, 118, 157) # make up two vectors of points
y <- c(181, 672, 175, 385, 529) # for capital cities

names <- c("London", "Edinburgh", "Cardiff", "Dublin", "Belfast") # vector of names

points(x,y, col="red") # add in the points in red

text(x + 20, y, labels=names, adj=0)
# add some labels displaced by 20km to the right

set.mine <- set.British.Isles$Object[c(1,2,3,87,88)]
blightly(place="set.mine", set=FALSE)
# specify a custom set of objects here it can
# be all the coastlines for the main British Isles
# note how you have to specify the new set in quotes
# and specify set=FALSE - all else should be as normal

blightly(parcol="gray") # use a basic shading

blightly("set.Scotland", parcol="blue", parbor="red", sarcol="red", tlncol="green", tlmwdh=3)
# looks hideous but demonstates some of the shading
# functions

blightly(parang=45, parden=30, parbor="transparent", parcol="black", sarbor="transparent", tlncol="transparent")
map.scale

# more of a picture for logos etc

---

**map.scale**  
**British Isles Coastline map scale**

**Description**

Function for drawing a simple scale on a blighty map

**Usage**

map.scale(pos="AUTO", width=1)

**Arguments**

- **pos**
  
  Can be AUTO (default), select, or a vector with the coordinates for the *centre* of the scale.

- **width**

  Integer value for the width of the bar.

**Value**

No return value

**Note**

This function draws a simple bar for a scale with the length of the bar in km beneath it. AUTO (default) draws the scale in the North-East corner of the map. select prompts for locator to allow the user to place the *centre* of the scale manually. If you send a vector then the vector gives the position of the *centre* of the scale.

You can’t use this with other high level plotting commands (such as oz) because blighty() stores the mapping parameters from it’s last invocation in an object called blighty.mapinfo from which the position, and size, of the scale is calculated.

**Author(s)**

David Lucy <<d.lucy@lancaster.ac.uk>> [http://www.maths.lancs.ac.uk/~lucy/](http://www.maths.lancs.ac.uk/~lucy/)

**See Also**

blighty north pointer oz

**Examples**

blighty() # draw coastline
map.scale(pos="AUTO", width=2) # automatic placing
map.scale(width=2) # same as above
## Not run: map.scale(pos="select") # locator to select
map.scale(pos=c(400,400)) # position vector
north.pointer  

British Isles Coastline North pointer

Description
Function for drawing a north pointer on a blighty map

Usage
north pointer(pos="AUTO")

Arguments
pos  Can be AUTO, select, or a vector with the coords. AUTO places the North pointer in the North-East corner of the map. select allows the user to select the centre of the N manually. Otherwise a vector of coordinates can be sent.

Value
No return value

Note
Draws a simple North pointer on a map. The position refers to the centre of the N on the pointer.
You can’t use this with other high level plotting commands (such as oz) because blighty() stores the mapping parameters from it’s last invocation in an object called blighty.mapinfo from which the position, and size, of the scale is calculated.

Author(s)
David Lucy <<d.lucy@lancaster.ac.uk>> http://www.maths.lancs.ac.uk/~lucy/

See Also
map.scale blighty oz

Examples
blighty() # draw coastline
north pointer(pos="AUTO") # automatic placing
north pointer() # same as above
## Not run: north pointer(pos="select") # uses locator to find the position
north pointer(pos=c(500,500)) # vector to determine the centre of the N of the pointer
Index

*Topic hplot
  blighty, 1
  map.scale, 5
  north.pointer, 6

blighty, 1, 5, 6
map.scale, 2, 4, 5, 6
north.pointer, 2, 4, 5, 6
oz, 3–6
points, 3
polygon, 2, 3