

Package ‘prettymapr’

August 29, 2016

Type Package

Title Scale Bar, North Arrow, and Pretty Margins in R

Version 0.1.5

Date 2016-04-05

Author Dewey Dunnington <dewey@fishandwhistle.net>

Maintainer Dewey Dunnington <dewey@fishandwhistle.net>

Description Automates the process of creating a scale bar and north arrow in any package that uses base graphics to plot in R. Bounding box tools help find and manipulate extents. Finally, there is a function to automate the process of setting margins, plotting the map, scale bar, and north arrow, and resetting graphic parameters upon completion.

License GPL-2

LazyData TRUE

Imports sp, rgdal, digest, rjson, foreach

Suggests maptools, raster, rosm

URL <https://github.com/paleolimbot/prettymapr>

BugReports <https://github.com/paleolimbot/prettymapr/issues>

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-04-09 00:37:54

R topics documented:

prettymapr-package	2
addnortharrow	3
addscalebar	4
geocode	5
makebbox	6
mergebbox	7

plotscalebar	8
prettymap	8
scalebarparams	10
searchbbox	11
zoombbox	12

Index	14
--------------	-----------

prettymapr-package *Scale Bar, North Arrow, And Pretty Margins In R*

Description

Contains functions to automatically plot north arrows and scalebars with minimal effort. Bounding box tools query online sources for lat/lon bounding box of human-readable names.

Details

This package automates the process of creating a scale bar ([addscalebar](#)) and north arrow ([addnortharrow](#)) on plots created by any package that uses base graphics to plot in R. Bounding box tools [searchbbox](#) and [zoombbox](#) help find and manipulate extents. Finally [prettymap](#) automates the process of setting margins, plotting the map, scalebar, and north arrow, and resetting graphic parameters upon completion.

Author(s)

Dewey Dunnington <dewey@fishandwhistle.net>

Examples

```
#dont test to cut down on check time

#bbox functions make it easy to manipulate bounding boxes
wolfville <- searchbbox("wolfville ns")
wolfvillezoomedout <- zoombbox(wolfville, 0.5)

#scalebar() makes it easy to add a scalebar to any map
library(maptools)
data(wrld_simpl)
plot(wrld_simpl, xlim=c(-66.86, -59.75), ylim=c(43, 47.3)) #Nova Scotia
addscalebar()

#also works in non-lat/lon coordinate systems
addscalebar(plotepsg=3395) #specify plot is in mercator projection
addscalebar(plotepsg=26920) #specify plot is in UTM Zone 20N

#addnortharrow() adds a north arrow
addnortharrow()

#prettymap() sets margins and plots scalebar and north arrow
```

```

library(maptools)
data(wrld_simpl)
prettyplot(plot(wrld_simpl, xlim=c(-66.86, -59.75), ylim=c(43, 47.3)),
           arrow.scale=1.1)

#example using {rosm} (open street map tiles)
library(rosm)
prettyplot(osm.plot(wolfville))

```

addnortharrow

Plot North Arrow

Description

Plot a north arrow (pointing directly "up") positioned based on current plot extents.

Usage

```

addnortharrow(pos = "topright", padin = c(0.15, 0.15), scale = 1,
             lwd = 1, border = "black", cols = c("white", "black"),
             text.col = "black")

```

Arguments

pos	Where to align the north arrow. One of "bottomleft", "bottomright", "topleft", or "topright".
padin	A vector of length 2 determining the distance in inches between the scalebar and the edge of the plottable area.
scale	Scale the default north arrow to make it bigger or smaller
lwd	The line width outlining the north arrow
border	The line color outlining the north arrow
cols	A vector of length 2 determining the two colors to be drawn for the north arrow
text.col	Color of the "N"

Examples

```

library(maptools)
data(wrld_simpl)
plot(wrld_simpl)
addnortharrow()

plot(1:5, 1:5, asp=1)
addnortharrow()

```

 addscalebar

Auto Plot Scalebar

Description

Automatically determines the geographical scale of the plot and draws a labelled scalebar.

Usage

```
addscalebar(plotunit = NULL, plotepsg = NULL, widthhint = 0.25,
  unitcategory = "metric", htin = 0.1, padin = c(0.15, 0.15),
  style = "bar", bar.cols = c("black", "white"), lwd = 1,
  linecol = "black", tick.cex = 0.7, labelpadin = 0.08, label.cex = 0.8,
  label.col = "black", pos = "bottomleft")
```

Arguments

plotunit	The unit which the current plot is plotted in, one of cm, m, km, in, ft, mi. or latlon. This parameter is optional if plotepsg is passed.
plotepsg	The projection of the current plot. If extents are valid lat/lons, the projection is assumed to be lat/lon (EPSG:4326), or Spherical Mercator otherwise (EPSG:3857). This is done to work seamlessly with OpenStreetMap packages.
widthhint	The fraction of the plottable width which the scale bar should (mostly) occupy.
unitcategory	One of "metric" or "imperial"
htin	Height (in inches) of the desired scale bar
padin	A vector of length 2 determining the distance in inches between the scalebar and the edge of the plottable area.
style	One of "bar" or "ticks".
bar.cols	If style=="bar", the colors to be repeated to make the bar.
lwd	The line width to use when drawing the scalebar
linecol	The line color to use when drawing the scalebar
tick.cex	If style=="ticks", the height of interior ticks.
labelpadin	The distance between the end of the scalebar and the label (inches)
label.cex	The font size of the label
label.col	The color of the label
pos	Where to align the scalebar. One of "bottomleft", "bottomright", "topleft", or "topright".

Examples

```

plot(1:5, 1:5, asp=1)
addscalebar(plotunit="m")

library(maptools)
data(wrld_simpl)
plot(wrld_simpl, xlim=c(-66.86, -59.75), ylim=c(43, 47.3)) #Nova Scotia
addscalebar()

#also works in non-lat/lon coordinate systems
addscalebar(plotepsg=3395) #specify plot is in mercator projection
addscalebar(plotepsg=26920) #specify plot is in UTM Zone 20N

```

geocode

Geocode Locations

Description

Geocode locations using the [Google Web API](#) or the [PickPoint.io API](#). Implemented from the `ggmap:geocode` function from the `ggmap` package (<https://cran.r-project.org/package=ggmap>) by David Kahle

Usage

```

geocode(location, output = c("data.frame", "list"), source = "default",
         messaging = FALSE, limit = 1, key = NULL, sensor = FALSE, ...)

```

Arguments

<code>location</code>	A character vector of locations to pass to the geocoding API.
<code>output</code>	One of <code>data.frame</code> or <code>list</code> . If <code>data.frame</code> , the results are distilled into columns: <code>query</code> , <code>source</code> , <code>status</code> , <code>rank</code> , <code>address</code> , <code>lon</code> , <code>lat</code> , <code>bbox_n</code> , <code>bbox_e</code> , <code>bbox_s</code> , <code>bbox_w</code> , and <code>id</code> . If <code>list</code> , the raw JSON output from the geocoding API is returned as a <code>list</code> (containing lists). The output of a failed geocode return will always have a <code>\$status</code> attribute describing the failure.
<code>source</code>	One of <code>"default"</code> , <code>"google"</code> or <code>"pickpoint"</code> . If <code>"default"</code> , the function calls <code>options("prettymapr.geosour")</code> or chooses <code>"pickpoint"</code> if none is set. If using <code>"pickpoint"</code> , please sign up for your own (free) API key to avoid using the default excessively.
<code>messaging</code>	TRUE if verbose messaging is desired.
<code>limit</code>	The number of results to return per query. This refers to individual locations, for which ambiguous queries may return multiple results (e.g. Halifax, Nova Scotia; Halifax, United Kingdom, etc.). The default is 1. Pass 0 if no limit on queries is desired.

key	API key if source="pickpoint".
sensor	TRUE if the location is generated from a sensor.
...	A number of key/value pairs to append to the URL, specifying further options specific to each API. Google users may wish to provide client and signature arguments for use with the enterprise version with the API, or specify additional constraints on geocoding.

Value

A list or data.frame; see documentation for output argument.

Examples

```
#don't test to speed up checking time

geocode("wolfville, ns")
geocode("wolfville, ns", output="list")
geocode("halifax", limit=0)
geocode("Paddy's Pub Wolfville NS", source="google")
geocode(c("Houston, TX", "San Antonio TX", "Cleavland OH"), source="google")

#fails quietly
geocode("don't even think about geocoding this")
geocode("don't even think about geocoding this", output="list")
```

makebbox

Create a Bounding Box

Description

Convenience method to create a bounding box like that returned by `sp::bbox()`. To generate a bounding box from lists of lat/lon values use `sp::bbox(cbind(lons, lats))`.

Usage

```
makebbox(n, e, s, w)
```

Arguments

n	North bounding latitude
e	East bounding longitude
s	South bounding latitude
w	West bounding longitude

Value

A 2x2 matrix describing a bounding box like that returned by `sp::bbox()`

See Also

`sp::bbox`

Examples

```
makebbox(45.125, -64.25, 44.875, -64.75)
```

mergebbox

Combine bounding boxes

Description

Create a single bounding box that encloses each bounding box.

Usage

```
mergebbox(...)
```

Arguments

... An arbitrary number of bounding boxes as generated by `sp::bbox`, [makebbox](#) or [searchbbox](#)

Value

A single bounding box that contains all of its arguments.

Examples

```
library(prettymapr)
box1 <- searchbbox("wolfville, ns")
box2 <- searchbbox("halifax, ns")
box3 <- searchbbox("kentville, ns")
mergebbox(box1, box2, box3)
```

plotscalebar

Raw Plot Scale Bar

Description

Just in case anybody is hoping to draw a custom scalebar, this is the method used to plot it. If you don't know what this is, you should probably be using [addscalebar](#).

Usage

```
plotscalebar(x, y, ht, params, style = "bar", adj = c(0, 0),
  tick.cex = 0.7, bar.cols = c("black", "white"), lwd = 1,
  linecol = "black")
```

Arguments

x	The position (user) to draw the scale bar
y	The position (user) to draw the scale bar
ht	The height(in user coordinates) to draw the scale bar
params	Scalebar parameters as generated by scalebarparams
style	One of bar or ticks
adj	Where to align the scale bar relative to x and y
tick.cex	If style=="ticks", the height of interior ticks.
bar.cols	A vector of color names to be repeated for a bar style scalebar.
lwd	Passed when drawing lines associated with the scalebar
linecol	Passed when drawing lines associated with the scalebar

See Also

[addscalebar](#)

prettymap

Plot A Pretty Map

Description

This function executes everything in plotexpression, then draws north arrow and scale bar using [addnortharrow](#) and [addscalebar](#). Specify that plot is in a non lat/lon projection by passing scale.plotepsg=... or plotunit="m".

Usage

```
prettymap(plotexpression, oma = c(0, 0, 0, 0), mai = c(0, 0, 0, 0),
  drawbox = FALSE, box.lwd = 1, drawscale = TRUE,
  scale.pos = "bottomleft", scale.htin = 0.1, scale.widthhint = 0.25,
  scale.unitcategory = "metric", scale.style = "bar",
  scale.bar.cols = c("black", "white"), scale.lwd = 1,
  scale.linecol = "black", scale.padin = c(0.15, 0.15),
  scale.labelpadin = 0.08, scale.label.cex = 0.8,
  scale.label.col = "black", scale.plotunit = NULL, scale.plotepsg = NULL,
  scale.tick.cex = 0.8, drawarrow = FALSE, arrow.pos = "topright",
  arrow.scale = 1, arrow.padin = c(0.15, 0.15), arrow.lwd = 1,
  arrow.cols = c("white", "black"), arrow.border = "black",
  arrow.text.col = "black", title = NULL, ...)
```

Arguments

plotexpression	An expression to plot the map, can be in brackets. e.g. plot(stuff); text(places, "readme!") or {plot(stuff); text(places, "readme!")}
oma	A vector of length 4 describing the outer margin area. See documentation for graphics::par.
mai	A vector of length 4 describing the margin area in inches. See documentation for graphics::par.
drawbox	TRUE if box should be drawn around map, FALSE otherwise.
box.lwd	The line width of the box
drawscale	TRUE if scalebar should be drawn, FALSE otherwise.
scale.pos	Where to align the scalebar. One of "bottomleft", "bottomright", "topleft", or "topright".
scale.htin	Height (in inches) of the desired scale bar
scale.widthhint	The fraction of the plottable width which the scale bar should (mostly) occupy.
scale.unitcategory	One of "metric" or "imperial"
scale.style	One of "bar" or "ticks".
scale.bar.cols	If style=="bar", the colors to be repeated to make the bar.
scale.lwd	The line width to use when drawing the scalebar
scale.linecol	The line color to use when drawing the scalebar
scale.padin	A vector of length 2 determining the distance in inches between the scalebar and the edge of the plottable area.
scale.labelpadin	The distance between the end of the scalebar and the label (inches)
scale.label.cex	The font size of the label
scale.label.col	The color of the label

<code>scale.plotunit</code>	The unit which the current plot is plotted in, one of cm, m, km, in, ft, mi. or latlon. This parameter is optional if <code>plotepsg</code> is passed.
<code>scale.plotepsg</code>	The projection of the current plot. If extents are valid lat/lons, the projection is assumed to be lat/lon (EPSG:4326), or Spherical Mercator otherwise (EPSG:3857). This is done to work seamlessly with OpenStreetMap packages.
<code>scale.tick.cex</code>	If <code>style=="ticks"</code> , the height of interior ticks.
<code>drawarrow</code>	TRUE if north arrow should be drawn, FALSE otherwise
<code>arrow.pos</code>	Where to align the north arrow. One of "bottomleft", "bottomright", "topleft", or "topright".
<code>arrow.scale</code>	Scale the default north arrow to make it bigger or smaller
<code>arrow.padin</code>	A vector of length 2 determining the distance in inches between the scalebar and the edge of the plottable area.
<code>arrow.lwd</code>	The line width outlining the north arrow
<code>arrow.cols</code>	A vector of length 2 determining the two colors to be drawn for the north arrow
<code>arrow.border</code>	The line color outlining the north arrow
<code>arrow.text.col</code>	Color of the "N"
<code>title</code>	Plot title, or NULL if none is desired.
<code>...</code>	Further graphical parameters to set while executing plotting code

Examples

```

prettymap(plot(1:5, 1:5, asp=1), scale.plotunit="cm", drawarrow=FALSE)
#add a title
prettymap(plot(1:5, 1:5, asp=1), title="My Plot")

library(maptools)
data(wrld_simpl)
prettymap({plot(wrld_simpl, xlim=c(-66.86, -59.75), ylim=c(43, 47.3))
  text(-62, 44, "Nova Scotia")
  text(-63, 47, "PEI")}, arrow.scale=1.1)

#also works in non-lat/lon coordinate systems
prettymap(plot(1:1000, 1:1000, asp=1),
  scale.plotepsg=26920, drawarrow=FALSE) #specify plot is in UTM Zone 20N

```

scalebarparams

Get Scale Bar Parameters

Description

Get default scale bar parameters based on the current plot (i.e. `par("usr")`). The algorithm attempts to detect the best equally divisible distance to use for the scale bar, and returns a list object with attributes that allow any type of scale bar to be drawn. The only way to manipulate the values chosen by the algorithm is to change the `widthhint` argument. For generic XY plots, pass `plotunit`.

Usage

```
scalebarparams(plotunit = NULL, plotepsg = NULL, widthhint = 0.25,
               unitcategory = "metric")
```

Arguments

plotunit	The unit which the current plot is plotted in, one of cm, m, km, in, ft, mi. or latlon. This parameter is optional if plotepsg is passed.
plotepsg	The projection of the current plot. If extents are valid lat/lons, the projection is assumed to be lat/lon (EPSG:4326), or Spherical Mercator otherwise (EPSG:3857). This is done to work seamlessly with OpenStreetMap packages.
widthhint	The fraction of the plottable width which the scale bar should (mostly) occupy.
unitcategory	One of "metric" or "imperial"

Value

a list of parameters: \$widthu (width of the scalebar in human readable units); \$unit (the human readable unit); \$majordivu (the size of the divisions in human readable units); \$majordivs (the number of divisions); \$widthplotunit (width of the scalebar in plotting units); \$majordivplotunit (the width of divisions in plotting units); \$labeltext (label text); and extents the user extents (par('usr')) that were used to calculate the parameters.

See Also

[addscalebar](#)

Examples

```
plot(1:5, 1:5, asp=1)
scalebarparams(plotunit="m")

library(maptools)
data(wrld_simpl)
plot(wrld_simpl, xlim=c(-66.86, -59.75), ylim=c(43, 47.3)) #Nova Scotia
scalebarparams()
```

searchbbox

Query The Interwebs For A Bounding Box

Description

Use the [PickPoint.io API](#) or Google API to retrieve a bounding box for the given query. Note that if you would like to use google as a source, you must agree to the Google API [terms and conditions](#).

Usage

```
searchbbox(querystring, ...)
```

Arguments

`querystring` The search query. Passing a vector in will find the bounding box that contains all bounding boxes returned.

`...` Additional paramters to be passed on to [geocode](#). Passing `source="google"` may be useful if google is desired as a source. Use `options(prettymapr.geosource="google")` to permanently use google as a source.

Value

A 2x2 matrix describing a bounding box like that returned by `sp::bbox()`

Examples

```
#don't test to speed up checking time

searchbbox("kings county, NS")
searchbbox("University Ave. Wolfville NS", source="google")
searchbbox("Wolfville ns", source="google")
searchbbox(c("Vermont", "Nova Scotia"))
```

 zoombbox

Zoom the extents of a bounding box

Description

Manipulate the extents of a bounding box by zooming and moving an existing bbox. This is helpful when manipulating the extents of a plot created by `canvec.qplot()`

Usage

```
zoombbox(bbox, factor = 1, offset = c(0, 0))
```

Arguments

`bbox` An existing bbox

`factor` A factor to zoom by. >1 will zoom in, <1 will zoom out. If a vector is passed, the first element will zoom the X extent, the second element will zoom the Y extent.

`offset` A vector describing the X and Y offset that should be applied.

Value

A zoomed bounding box.

Examples

```
alta <- searchbbox("alta lake bc", source="google")
zoombbox(alta, c(.2,.5))
```

Index

*Topic **package**

prettymapr-package, 2

addnortharrow, 2, 3, 8

addscalebar, 2, 4, 8, 11

geocode, 5, 12

makebbox, 6, 7

mergebbox, 7

plotscalebar, 8

prettymap, 2, 8

prettymapr (prettymapr-package), 2

prettymapr-package, 2

scalebarparams, 8, 10

searchbbox, 2, 7, 11

zoombbox, 2, 12