Package 'OpenStreetMap'

September 9, 2016

Maintainer Ian Fellows <ian@fellstat.com>

License GPL-2 | file LICENCE

Title Access to Open Street Map Raster Images

Author Ian Fellows, using the JMapViewer library by Jan Peter Stotz

Description Accesses high resolution raster maps using the OpenStreetMap protocol. Dozens of road, satellite, and topographic map servers are directly supported, including Apple, Mapnik, Bing, and stamen. Additionally raster maps may be constructed using custom tile servers. Maps can be plotted using either base graphics, or ggplot2. This package is not affiliated with the OpenStreetMap.org mapping project.

SystemRequirements Java (>= 1.5), JRI

Version 0.3.3

URL http://www.fellstat.com http://blog.fellstat.com/?cat=15

Date 2016-09-09

Depends methods, R (>= 2.10.0)

Suggests colorspace, maptools

Imports grDevices, ggplot2 (>= 0.9.0), rJava, raster, rgdal, sp

Collate 'OpenStreetMap-package.R' 'osm.R' 'autoplot.R' 'zzz.R'

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-09-09 23:49:52

R topics documented:

`autoplot.OpenStreetMap` *Plot an open street map using ggplot2*

Description

Plot an open street map using ggplot2

Usage

```
## S3 method for class 'OpenStreetMap'
autoplot(data, expand = TRUE, ...)
```

Arguments

`data`	an OpenStreetMap object
`expand`	if true the plotting bounds are expanded to the bounding box
`...`	not used

Examples

```
## Not run:
require(maps)
require(ggplot2)

mp <- openmap(c(53.38332836757155,-130.517578125),
c(15.792253570362446,-67.939453125),4,'stamen-watercolor')
mp_bing <- openmap(c(53.38332836757155,-130.517578125),
c(15.792253570362446,-67.939453125),4,'bing')
states_map <- map_data("state")
states_map_merc <- as.data.frame(
projectMercator(states_map$lat,states_map$long))
states_map_merc$region <- states_map$region
states_map_merc$group <- states_map$group
```

```
crimes <- data.frame(state = tolower(rownames(USArrests)), USArrests)

p <- autoplot(mp,expand=FALSE) + geom_polygon(aes(x=x,y=y,group=group),
data=states_map_merc,fill="black",colour="black",alpha=.1) + theme_bw()
print(p)
p <- autoplot(mp_bing) + geom_map(aes(x=-10000000,y=4000000,map_id=state,fill=Murder),
data=crimes,map=states_map_merc)
print(p)

## End(Not run)
```

`autoplot.osmtile` *Plots an open street map tile using ggplot2*

Description

Plots an open street map tile using ggplot2

Usage

```
## S3 method for class 'osmtile'
autoplot(data, plot = FALSE, ...)
```

Arguments

`data`	an osmtile
`plot`	if false only the annotation_raster is returned
`...`	not used

`getMapInfo` *Returns a table with relevant source and attribution info for each map type*

Description

Returns a table with relevant source and attribution info for each map type

Usage

```
getMapInfo()
```

launchMapHelper *Launches a Java helper GUI.*

Description

Launches a Java helper GUI.

Usage

```
launchMapHelper()
```

Details

note for Mac OS X users: On the mac this can only be run from a java console such as JGR.

LA_places *Places of interest in Los Angeles*

Description

Places of interest in Los Angeles

longlat *Latitude Longitude projection*

Description

Latitude Longitude projection

Usage

```
longlat()
```

openmap *Get a map based on lat long coordinates*

Description

Get a map based on lat long coordinates

Usage

```
openmap(upperLeft, lowerRight, zoom = NULL, type = c("osm", "osm-bw",
  "maptoolkit-topo", "waze", "bing", "stamen-toner", "stamen-terrain",
  "stamen-watercolor", "osm-german", "osm-wanderreitkarte", "mapbox", "esri",
  "esri-topo", "nps", "apple-iphoto", "skobbler", "hillshade", "opencyclemap",
  "osm-transport", "osm-public-transport", "osm-bbike", "osm-bbike-german"),
  minNumTiles = 9L, mergeTiles = TRUE)
```

Arguments

upperLeft	the upper left lat and long
lowerRight	the lower right lat and long
zoom	the zoom level. If null, it is determined automatically
type	the tile server from which to get the map, or the url pattern.
minNumTiles	If zoom is null, zoom will be chosen such that the number of map tiles is greater than or equal to this number.
mergeTiles	should map tiles be merged into one tile

Details

Type may be the url of a custom tile server (http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification). should include z, y, and x specifying where the zoom, xtile and ytile location should be substituted. e.g.

http://api.someplace.com/.../z/x/y

Examples

```
## Not run:
#show some of the maps available
nm <- c("osm", "maptoolkit-topo", "bing", "stamen-toner",
"stamen-watercolor", "esri", "esri-topo",
"nps", "apple-iphoto", "skobbler")
par(mfrow=c(3,4))
#Korea
for(i in 1:length(nm)){
map <- openmap(c(43.46886761482925,119.94873046875),
c(33.22949814144951,133.9892578125),
minNumTiles=3,type=nm[i])
```

```
plot(map)
}
# Some maps from custom urls
apiKey <- paste0("?access_token=",
 "pk.eyJ1IjoidGhlZmVsbCIsImEiOiJjaXN1anNwODEwMWlrMnRvZHBhamRrZjlqIn0.Gf8qLSpZ6yo5yfQhEutFfQ")
baseUrl <- "https://api.mapbox.com/styles/v1/mapbox/satellite-streets-v9/tiles/256/{z}/{x}/{y}"
map <- openmap(c(43.46886761482925,119.94873046875),
c(33.22949814144951,133.9892578125),
minNumTiles=4,
type=paste0(baseUrl,apiKey))
plot(map)

baseUrl <- "https://api.mapbox.com/styles/v1/mapbox/dark-v9/tiles/256/{z}/{x}/{y}"
map <- openmap(c(43.46886761482925,119.94873046875),
c(33.22949814144951,133.9892578125),
minNumTiles=4,
type=paste0(baseUrl,apiKey))
plot(map)


#plot Korea with ggplot2.
library(ggplot2)
map <- openmap(c(43.46886761482925,119.94873046875),
c(33.22949814144951,133.9892578125),
minNumTiles=4)
autoplot(map)


## End(Not run)
```

`openproj` *Projects the open street map to an alternate coordinate system*

Description

Projects the open street map to an alternate coordinate system

Usage

```
openproj(x, projection = "+proj=longlat", ...)
```

Arguments

`x`	an OpenStreetMap object
`projection`	a proj4 character string or CRS object
`...`	additional parameters for projectRaster

Examples

```
## Not run:
library(maps)

#plot bing map in native mercator coords
map <- openmap(c(70,-179),
c(-70,179),zoom=1,type='bing')
plot(map)

#using longlat projection lets us combine with the maps library
map_longlat <- openproj(map)
plot(map_longlat)
map("world",col="red",add=TRUE)

#robinson projection. good for whole globe viewing.
map_robinson <- openproj(map_longlat, projection=
"+proj=robin +lon_0=0 +x_0=0 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m +no_defs")
plot(map_robinson)

#national parks service images
upperMap <- openmap(c(70,-179),
c(10,50),zoom=2,type='nps')
#Lambert Conic Conformal
map_llc <- openproj(upperMap, projection=
"+proj=lcc +lat_1=33 +lat_2=45 +lat_0=39 +lon_0=-96")
plot(map_llc,removeMargin=TRUE)
#add choropleth
library(sp)
data(states)
st_llc <- spTransform(states,CRS("+proj=lcc +lat_1=33 +lat_2=45 +lat_0=39 +lon_0=-96"))
plot(st_llc,add=T,col=heat.colors(48,.4)[slot(st_llc,"data")[["ORDER_ADM"]]])


## End(Not run)
```

`osm` *Open street map (and google) mercator projection*

Description

Open street map (and google) mercator projection

Usage

```
osm()
```

osmtile *Get an open street map tile.*

Description

Get an open street map tile.

Usage

```
osmtile(x, y, zoom, type = "osm")
```

Arguments

x	location in osm native coordinates
y	location in osm native coordinates
zoom	zoom level
type	the map type (see getMapInfo)

Value

a tile

plot.OpenStreetMap *Plot an OpenStreetMap object.*

Description

Plot an OpenStreetMap object.

Usage

```
## S3 method for class 'OpenStreetMap'
plot(x, y = NULL, add = FALSE,
  removeMargin = TRUE, ...)
```

Arguments

x	the OpenStreetMap
y	ignored
add	add to current plot
removeMargin	remove margins from plotting device
...	additional parameters to be passed to plot

Examples

```
## Not run:
#
# The following examples
# plot using native mercator coordinates,
# transforming the data where needed
#
library(sp)
m <- c(25.7738889,-80.1938889)
j <- c(58.3019444,-134.4197222)
miami <- projectMercator(25.7738889,-80.1938889)
jun <- projectMercator(58.3019444,-134.4197222)
data(states)
map <- openmap(j,m,4,type="stamen-terrain")
plot(map,removeMargin=FALSE)
plot(states,add=TRUE)

data(LA_places)
longBeachHarbor <- openmap(c(33.760525217369974,-118.22052955627441),
c(33.73290566922855,-118.17521095275879),14,'bing')
coords <- coordinates(LA_places)
x <- coords[,1]
y <- coords[,2]
txt <- slot(LA_places,"data")[,'NAME']
plot(longBeachHarbor)
points(x,y,col="red")
text(x,y,txt,col="white",adj=0)

if(require(UScensus2010)){
#install with: install.tract("linux")
if(require(UScensus2010tract)){
lat <- c(43.834526782236814,30.334953881988564)
lon <- c(-131.0888671875  ,-107.8857421875)
southwest <- openmap(c(lat[1],lon[1]),c(lat[2],lon[2]),5,'osm')
data(california.tract10)
cali <- spTransform(california.tract10,osm())

plot(southwest)
plot(cali,add=TRUE)
}
}

#
# The same plot using apple's maps and long-lat coordinates,
#   transforming the raster map.
#
if(require(UScensus2010)){
#install with: install.tract("linux")
if(require(UScensus2010tract)){
lat <- c(43.834526782236814,30.334953881988564)
lon <- c(-131.0888671875  ,-107.8857421875)
southwest <- openmap(c(lat[1],lon[1]),
```

```
c(lat[2],lon[2]),5,"apple-iphoto")
southwest_longlat <- openproj(southwest)
data(california.tract10)
plot(southwest_longlat)
plot(california.tract10,add=TRUE)
}
}

## End(Not run)
```

`plot.osmtile` *Add tile to plot*

Description

Add tile to plot

Usage

```
## S3 method for class 'osmtile'
plot(x, y = NULL, add = TRUE, raster = TRUE, ...)
```

Arguments

`x`	the tile
`y`	ignored
`add`	add to current plot (if raster, then image is always added)
`raster`	use raster image
`...`	additional parameters to image or rasterImage

`print.OpenStreetMap` *Print map*

Description

Print map

Usage

```
## S3 method for class 'OpenStreetMap'
print(x, ...)
```

Arguments

`x`	the OpenStreetMap
`...`	ignored

projectMercator *Maps long lat values to the open street map mercator projection*

Description

Maps long lat values to the open street map mercator projection

Usage

```
projectMercator(lat, long, drop = TRUE)
```

Arguments

lat	a vector of latitudes
long	a vector of longitudes
drop	drop to lowest dimension

raster,OpenStreetMap-method *Create a RasterLayer from an OpenStreetMap*

Description

Create a RasterLayer from an OpenStreetMap

Usage

```
## S4 method for signature 'OpenStreetMap'
raster(x, ...)
```

Arguments

x	an OpenStreetMap
...	unused

Examples

```
## Not run:
library(raster)
longBeachHarbor <- openmap(c(33.760525217369974,-118.22052955627441),
c(33.73290566922855,-118.17521095275879),14,'bing')
ras <- raster(longBeachHarbor)
plotRGB(ras)

## End(Not run)
```

raster,osmtile-method *Create a RasterLayer from a tile*

Description

Create a RasterLayer from a tile

Usage

```
## S4 method for signature 'osmtile'
raster(x, ...)
```

Arguments

x	an osmtile
...	unused

states *The United States*

Description

The United States

Index