

Package ‘OpenStreetMap’

July 2, 2014

Maintainer Ian Fellows <ian@fellstat.com>

License GPL-2 | file LICENCE

Title Access to open street map raster images

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

Description

Accesses high resolution raster maps. Dozens of road, satellite, and topographic map servers are directly supported, including CloudMade, Mapnik, Bing, stamen, and MapQuest. Maps can be plotted using either base graphics, or ggplot2.

SystemRequirements Java (>= 1.5), JRI

Version 0.3.1

URL <http://www.fellstat.com> <http://blog.fellstat.com/?cat=15>
<http://research.cens.ucla.edu/>

Date 2013-04-11

Depends rJava, methods, raster, rgdal, R (>= 2.10.0)

Suggests ggplot2 (>= 0.9.0), sp, colorspace, maptools

Imports rJava

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

NeedsCompilation no

Repository CRAN

Date/Publication 2013-04-13 20:38:13

R topics documented:

autoplot.OpenStreetMap	2
autoplot.osmtile	3
getMapInfo	4
launchMapHelper	4
LA_places	4
longlat	4
openmap	5
openproj	6
osm	7
osmtile	7
plot.OpenStreetMap	8
plot.osmtile	9
print.OpenStreetMap	10
projectMercator	10
raster	11
setCloudMadeKey	11
states	11
Index	12

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)
gpplibPermit()

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	<i>Returns a table with relevant source and attribution info for each map type</i>
------------	--

Description

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

Usage

```
getMapInfo()
```

launchMapHelper	<i>Launches a Java helper GUI.</i>
-----------------	------------------------------------

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	<i>Places of interest in Los Angeles</i>
-----------	--

Description

Places of interest in Los Angeles

longlat	<i>Latitude Longitude projection</i>
---------	--------------------------------------

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", "mapquest", "mapquest-aerial", "bing", "stamen", "nps", "apple-iphoto", "skobbler"),
        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
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

Examples

```
## Not run:
#show some of the maps available
nm <- c("osm", "maptoolkit-topo", "mapquest",
        "mapquest-aerial", "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)
}

#cloudMade has thousands of map types, and requires a key.
#A default key is provided with the package, but you
#should get your own at http://ww.cloudmade.com and
#apply it with:
#setCloudMadeKey("< your key >")

#plot Korea with a cloudmade map and ggplot2.
```

```
map <- openmap(c(43.46886761482925,119.94873046875),
c(33.22949814144951,133.9892578125),
minNumTiles=4,type="cloudmade-1960")
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 mapquest map in native mercator coords
map <- openmap(c(70,-179),
c(-70,179),zoom=1,type='mapquest-aerial')
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
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	<i>Open street map (and google) mercator projection</i>
-----	---

Description

Open street map (and google) mercator projection

Usage

```
osm()
```

osmtile	<i>Get an open street map tile.</i>
---------	-------------------------------------

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 <code>getMapInfo</code>)

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
#
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	<i>Create a RasterLayer from a tile</i>
--------	---

Description

Create a RasterLayer from a tile

Create a RasterLayer from an OpenStreetMap

Arguments

x	an osmtile
...	unused
x	an OpenStreetMap
...	unused

Examples

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

## End(Not run)
```

setCloudMadeKey	<i>Sets the user identification key for cloudmade.com</i>
-----------------	---

Description

Sets the user identification key for cloudmade.com

Usage

```
setCloudMadeKey(key)
```

Arguments

key	The key. Obtain a free map key at http://www.cloudmade.com
-----	---

states	<i>The United States</i>
--------	--------------------------

Description

The United States

Index

*Topic **datasets**

LA_places, [4](#)
states, [11](#)

*Topic **data**

LA_places, [4](#)
states, [11](#)

autoplot.OpenStreetMap, [2](#)
autoplot.osmtile, [3](#)

getMapInfo, [4](#)

LA_places, [4](#)
launchMapHelper, [4](#)
longlat, [4](#)

openmap, [5](#)
openproj, [6](#)
osm, [7](#)
osmtile, [7](#)

plot.OpenStreetMap, [8](#)
plot.osmtile, [9](#)
print.OpenStreetMap, [10](#)
projectMercator, [10](#)

raster, [11](#)

setCloudMadeKey, [11](#)
states, [11](#)