Package 'OpenStreetMap'

March 1, 2012

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 Access to high quality open street map raster images.

SystemRequirements Java (>= 1.5), JRI

Version 0.2

URL http://www.fellstat.com http://research.cens.ucla.edu/

Date 2012-02-21

Depends methods, rJava, sp, maptools, raster, R (>= 2.10.0)

Suggests rgdal, ggplot2 (>= 0.9.0), colorspace

Imports rJava

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

Repository CRAN

Date/Publication 2012-03-01 19:37:43

R topics documented:

`OpenStreetMap-package` *Open Street Maps.*

Description

Open street maps.

Details

Package:	OpenStreetMap
Type:	Package
Version:	1.0
Date:	2011-04-04
License:	LGPL-2
LazyLoad:	yes

Includes The ability to plot Open Street Map and Bing satellite images.

Author(s)

Ian Fellows <ian.fellows@stat.ucla.edu>

References

http://www.fellstat.com

`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)
require(rgdal)
gpclibPermit()

mp <- openmap(c(53.38332836757155,-130.517578125),
c(15.792253570362446,-67.939453125),4,'osm')
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

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 = "osm",
  minNumTiles = 9L)
```

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	'osm' for mapnik open street map, or 'bing' for bing aerial
minNumTiles	If zoom is null, zoom will be chosen such that the number of map tiles is greater than or equal to this number.

Examples

```
## Not run:
#Korea
map <- openmap(c(43.46886761482925,119.94873046875),
c(33.22949814144951,133.9892578125),type='osm')
plot(map,raster=TRUE)

## 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(rgdal)
library(maps)

#plot map in native mercator coords
map <- openmap(c(70,-179),
c(-70,179),zoom=2,type='bing')
plot(map)

#using longlat projection lets us combine with the maps library
```

```
map_longlat <- openproj(map)
plot(map_longlat,raster=TRUE)
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)


map <- openmap(c(70,-179),
c(40,179),zoom=2,type='bing')
map_longlat <- openproj(map)
#Lambert Conic Conformal (takes some time...)
map_llc <- openproj(map_longlat, projection=
"+proj=lcc +lat_1=33 +lat_2=45 +lat_0=39 +lon_0=-96")
plot(map_llc,raster=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` *open street map (and google) mercator projection*

Description

open street map (and google) mercator projection

Usage

```
osm()
```

`osmtile` *get an open street map tile. tpe can be "osm" or "bing"*

Description

get an open street map tile. tpe can be "osm" or "bing"

Usage

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

Arguments

x	location in osm native coordinates
y	location in osm native coordinates
zoom	zoom level
type	osm for mapnik open street map, or 'bing' for bing aerial

Value

a tile

---

plot.OpenStreetMap *plot the map in mercator coordinates. see osm().*

Description

plot the map in mercator coordinates. see osm().

Usage

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

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:
library(rgdal)
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)
plot(map,removeMargin=TRUE)
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,removeMargins=TRUE,raster=TRUE)
points(x,y,col="red")
text(x,y,txt,col="white",adj=0)

 library(UScensus2000)

 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.tract)
 california.tract <- spTransform(california.tract,osm())

 plot(southwest,removeMargin=TRUE)
 plot(california.tract,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 = FALSE, ...)
```

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 *create a RasterLayer from a tile*

Description

create a RasterLayer from a tile

create a RasterLayer from an OpenStreetMap

Arguments

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

states *The United States*

Description

The United States

Index