

Package: quickmapr (via r-universe)

November 5, 2024

Type Package

Title Quickly Map and Explore Spatial Data

Version 0.4.0

URL <https://github.com/geogugal/quickmapr>

BugReports <https://github.com/geogugal/quickmapr/issues>

Maintainer Jeffrey W. Hollister <hollister.jeff@epa.gov>

License CC0

Description While analyzing geospatial data, easy visualization is often needed that allows for quick plotting, and simple, but easy interactivity. Additionally, visualizing geospatial data in projected coordinates is also desirable. The 'quickmapr' package provides a simple method to visualize 'sp', 'sf' (via coercion to 'sp'), and 'raster' objects, allows for basic zooming, panning, identifying, labeling, selecting, and measuring spatial objects. Importantly, it does not require that the data be in geographic coordinates.

Depends R (>= 3.0.0)

Imports raster, sp, httr, stats, grDevices, graphics, methods, sf

Suggests testthat, knitr

RoxygenNote 7.2.3

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev libssl-dev
libproj-dev libsqlite3-dev libudunits2-dev

Repository <https://jhollist.r-universe.dev>

RemoteUrl <https://github.com/geogugal/quickmapr>

RemoteRef HEAD

RemoteSha 142c9b29012a4f735f4767b4b51d668c7a576e84

Contents

buffer	2
elev	2
examp_lakes	3
f	3
i	4
l	4
lake	5
length	5
m	6
nlcd	6
p	6
plot.qmap	7
print.qmap	7
qmap	8
quickmapr	9
s	9
samples	10
width	10
ze	11
zi	11
zo	12

Index 13

buffer	<i>Buffer of Lake Sunapee for quickmapr examples</i>
--------	--

Description

This example data is a SpatialPolygonsDataFrame representing a buffer of a single lake, Lake Sunapee.

Format

SpatialPolygonsDataFrame

elev	<i>Elevation Data for quickmapr examples</i>
------	--

Description

This example data is a RasterLayer of the National Elevation Dataset for a buffer of Lake Sunapee.

Format

RasterLayer of 455 x 246 and 30 meter resolution

examp_lakes	<i>Lakes from the National Lakes Morphometry Dataset for quickmapr examples</i>
-------------	---

Description

This example data is a polygon dataset of many lakes in New Hampshire. Data for each lake are from the first version of the [National Lakes Morphometry Dataset](#).

Format

SpatialPolygonsDataFrame

f	<i>Returns qmap object to original extent</i>
---	---

Description

Allows for restoring map to original extent. The initial map is replayed and the map extent of the object is reset to match the current display.

Usage

f(qmap_obj = NULL)

Arguments

qmap_obj a qmap_obj to restore

Examples

```
## Not run:
data(lake)
x<-qmap(list(lake,buffer,elev))
zi(x)
f(x)

## End(Not run)
```

i	<i>Identify</i>
---	-----------------

Description

Interactively select an sp or raster object and print the data associated with it. `i()` only accepts a single input point.

Usage

```
i(qmap_obj = NULL, i_idx = 1, loc = NULL)
```

Arguments

<code>qmap_obj</code>	a qmap object from which to identify features.
<code>i_idx</code>	a numeric value specifying which data layer to identify or a character string indicating the name of the layer. Defaults to 1.
<code>loc</code>	A list with an x and y numeric indicating a location. Default is to interactively get loc value until escaped.

Value

Returns NULL. Identified values are printed to the screen.

Examples

```
## Not run:
data(lake)
qm<-qmap(list(lake,elev,samples))
i(qm,"lake")
i(qm,"samples")
i(qm,2)

## End(Not run)
```

1	<i>Label features</i>
---	-----------------------

Description

It is useful to be able to provide labels to features on your map when examining the results of given analysis. This function adds labels for a given layer (currently just point or polygon layers).

Usage

```
l(qmap_obj, field = NULL, layer = 1)
```

Arguments

qmap_obj	a qmap object from which to pull the labels. Raster layers are ignored. Will also accept sp objects.
field	a field in the sp object to use to label the features. Defaults to row.names().
layer	identify which sp layer to label. Defaults to first layer in qmap_obj\$map_data.

Examples

```
## Not run:
data(lake)
qm<-qmap(lake,width,buffer)
l(qm, 'COMID')
l(qm, layer = 2)
l(qm, layer = "buffer")

## End(Not run)
```

lake

Lake Data for quickmapr examples

Description

This example data is a SpatialPolygonsDataFrame of a single lake, Lake Sunapee.

Format

SpatialPolygonDataframe with 1 lakes, each with 13 variables

length

Maximum Lake Length line for quickmapr examples

Description

This example data is a line dataset representing the maximum lake length.

Format

SpatialLines

m	<i>Measure</i>
---	----------------

Description

Allows interactive selection of points and returns length of selected line in units of the current qmap object.

Usage

```
m(qmap_obj = NULL)
```

Arguments

qmap_obj a qmap object from which to measure features.

nlcd	<i>Land Cover Data for quickmapr examples</i>
------	---

Description

This example data is a RasterLayer of the National Land Cover Dataset for a buffer of Lake Sunapee.

Format

RasterLayer of 467 x 262 and 30 meter resolution

p	<i>Pan the current plot</i>
---	-----------------------------

Description

Interactively reposition the current plot. Works on an existing qmap object. Simply pass that object to p(). A single repositioning results and the extent of the qmap object is changed.

Usage

```
p(qmap_obj = NULL, loc = NULL)
```

Arguments

qmap_obj A qmap object. Optional, but performs better with larger data sets.
 loc A list with an x and y numeric indicating a location. Default is to interactively get loc value until escaped.

Examples

```
## Not run:  
data(lake)  
x<-qmap(list(lake,buffer,elev))  
p()  
## Or  
p(x)  
  
## End(Not run)
```

plot.qmap *Default plotting of a qmap object*

Description

Plots the qmap class and uses the order, colors, extent, and fill option from qmap.

Usage

```
## S3 method for class 'qmap'  
plot(x, ...)
```

Arguments

x input qmap class to plotS
... options passed to image or plot

print.qmap *Default plotting of a qmap object*

Description

Plots a qmap object

Usage

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

Arguments

x input qmap class to print
... options passed to plot

qmap

*Build qmap object***Description**

This function builds the qmap object that forms the basis for the rest of the [quickmapr](#) package.

Usage

```
qmap(
  ...,
  extent = NULL,
  order = 1:length(mapdata),
  colors = 1:length(mapdata),
  fill = FALSE,
  prj = TRUE,
  basemap = c("none", "1m_aerial", "topo"),
  resolution = 300
)
```

Arguments

...	Spatial objects to map. Maybe passed as objects, a list of spatial objects, or a qmap object
extent	A sp or raster object to use as the initial extent of the map. Defaults to the maximum extent of all input object
order	draw order of the spatial object. Defaults to order in mapdata
colors	line colors. Defaults to 1:length(mapdata)
fill	Logical to determine if polygons should be filled (using colors) or just the border colored.
prj	Logical to check projections of input spatial objects. Transformation, if needed, should be done prior to mapping with <code>sp::spTransform()</code> .
basemap	a basemap generated from get_basemap
resolution	Specifies the width in pixels of the retrieved basemap. Larger values result in higher resolution images but since the images are downloaded for each zoom level can result in delays. Default is 300, while ~600 is a decent compromise for performance and image quality.

Value

Function displays a map from the input mapdata parameter and returns a recorded plot.

Examples

```
## Not run:
data(lake)
mymap<-list(elev,lake,buffer,length,samples)
qm<-qmap(mymap)
#change draw order and which data is displayed
qmap(qm,order=c(2,3,5))
#add a basemap
qm<-qmap(qm,basemap="1m_aerial", resolution = 800)

## End(Not run)
```

quickmapr

Quick visualization of spatial data with simple interactivity

Description

This package provides tools to quickly visualize and interact with [sp](#) and [raster](#) objects. Simple interactivity includes panning, zooming, identifying, and labelling. The goal of [quickmapr](#) is to make working with spatial data easy and quick, it is not intended to provide publication quality maps.

s

Select

Description

Interactively select a single [sp](#) or [raster](#) object and return the object. `s()` only accepts a single input point.

Usage

```
s(qmap_obj = NULL, s_idx = 1, loc = NULL)
```

Arguments

qmap_obj	a qmap object from which to select features.
s_idx	a numeric value specifying which data layer to select or a character string indicating the name of the layer. Defaults to 1.
loc	A list with an x and y numeric indicating a location. Default is to interactively get loc value.

Value

Returns a selected [sp](#) object

Examples

```
## Not run:
data(lake)
qm<-qmap(list(lake,elev,samples))
s(qm,"lake")
s(qm,3)

## End(Not run)
```

samples	<i>Point data for quickmapr examples</i>
---------	--

Description

This is a fabricated point dataset for what might be a common type of data collected for a lake.

This example data is a fabricated point dataset with data for each location.

Format

SpatialPointsDataFrame

SpatialPointsDataFrame

width	<i>Maximum lake width for quickmapr examples</i>
-------	--

Description

This example data is a line dataset representing the maximum lake width. Maximum lake width is defined as the longest line perpendicular to the maximum lake length.

Format

SpatialLines

ze *Zooms in on extent*

Description

Select a bounding box interactively and zoom to that extent. Works on an existing qmap object. Simply pass that object to `ze()`. A single zoom in to the selected extent results and the extent of the qmap object is changed.

Usage

```
ze(qmap_obj = NULL, extent = NULL)
```

Arguments

`qmap_obj` A qmap object. Optional, but performs better with larger data sets.
`extent` A Spatial* object to specify extent to zoom into.

Examples

```
## Not run:
data(lake)
qmap(list(lake,buffer,elev))
ze()

## End(Not run)
```

zi *Zooms in on current plot*

Description

Interactively zoom in on the current plot. Works on an existing qmap object. Simply pass that object to `zi()`. A single zoom in results and the extent of the qmap object is changed.

Usage

```
zi(qmap_obj = NULL, zoom_perc = 0.5, loc = NULL)
```

Arguments

`qmap_obj` A qmap object. Optional, but performs better with larger data sets.
`zoom_perc` A proportion to determine the zoom level. The x and y axes are reduced by this amount. Default is 0.5.
`loc` A list with an x and y numeric indicating a location. Default is to interactively get loc value until escaped.

Examples

```
## Not run:
data(lake)
x<-qmap(list(lake,buffer,elev))
zi(x)

## End(Not run)
```

zo

Zooms out on current plot

Description

Interactively zoom out on the current plot. Works on an existing qmap object. Simply pass that object to `zo()`. A single zoom out results and the extent of the qmap object is changed.

Usage

```
zo(qmap_obj = NULL, zoom_perc = 0.5, loc = NULL)
```

Arguments

qmap_obj	A qmap object. Optional, but performs better with larger data sets.
zoom_perc	A proportion to determine the zoom level. The x and y axes are increased by this amount. Default is 0.5.
loc	A list with an x and y numeric indicating a location. Default is to interactively get loc value until escaped.

Examples

```
## Not run:
data(lake)
qm<-qmap(list(lake,buffer,elev))
zo(qm)

## End(Not run)
```

Index

- * **datasets**
 - buffer, 2
 - elev, 2
 - examp_lakes, 3
 - lake, 5
 - length, 5
 - nlcd, 6
 - samples, 10
 - width, 10
- width, 10
- ze, 11
- zi, 11
- zo, 12
- buffer, 2
- elev, 2
- examp_lakes, 3
- f, 3
- get_basemap, 8
- i, 4
- l, 4
- lake, 5
- length, 5
- m, 6
- nlcd, 6
- p, 6
- plot.qmap, 7
- print.qmap, 7
- qmap, 8, 8
- quickmapr, 8, 9, 9
- raster, 8, 9
- s, 9
- samples, 10
- sp, 8, 9