Web Maps

Barry Rowlingson

August 27, 2009

The `webmaps` package does two main things: converting spatial data from R to create HTML/Javascript maps using OpenLayers, and importing map tile data into R for use on R graphic plots.

1 Creating HTML maps

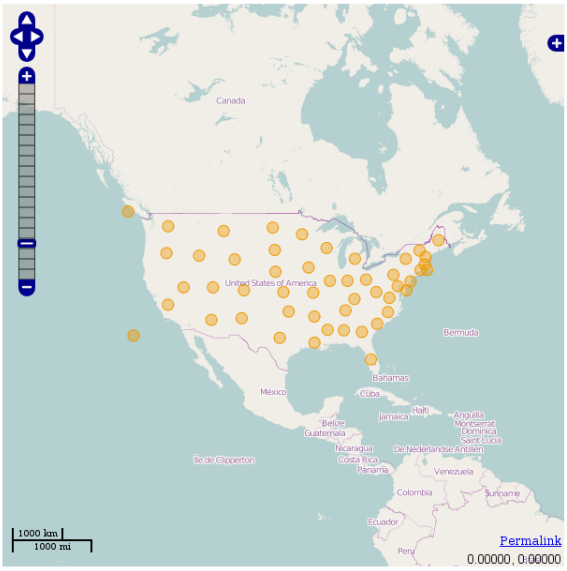
The `webmaps` package takes spatial data and produces an HTML page that overlays that data onto a map. It uses the OpenLayers javascript library and OpenStreetMap as base data. In this way it produces freely-usable maps for the web with no need to get an API key or submit to some restrictive re-use license.

Here is a simple example:

```
> library(webmaps)
> state = data.frame(state.x77)
> state$Name = rownames(state)
> coordinates(state) = cbind(state.center$x, state.center$y)
> osmMap(layer(state,"States"), title="State␣Data",outputDir ="./states1")
```

Pointing your web browser to the generated html file produces this:

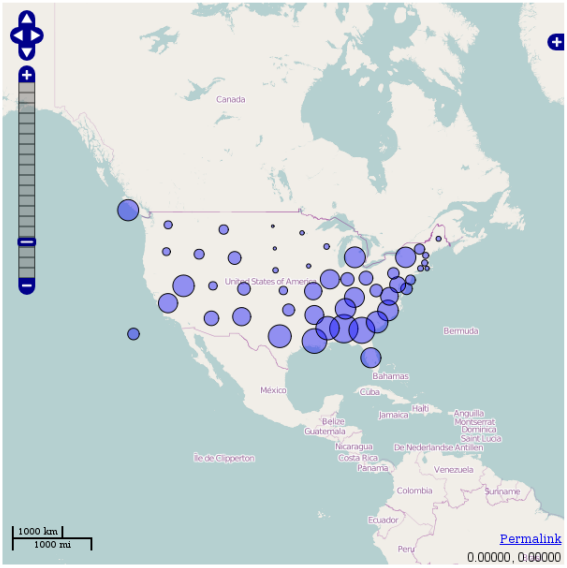
State Data



The package can produce maps with several layers of information containing points, polygons and lines. The appearance of layers can be individually controlled.

The appearance of individual features can also be set. Here we scale the size of the points by the murder rate in each state:

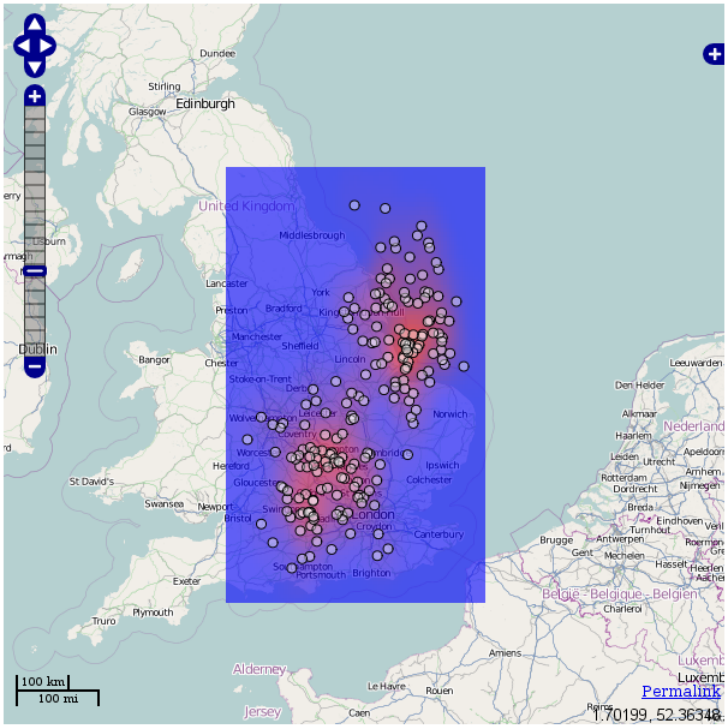
State Data



It's also possible to overlay raster data by using `ilayer`. Here we generate some points and use the `splancs` kernel smoothing function to produce a surface:

```
> library(splancs)
> pts = cbind(rnorm(100,0.5,.5),rnorm(100,53.5,.5))
> pts = rbind(pts,cbind(rnorm(100,-1,.5),rnorm(100,52,.5)))
> k = kernel2d(pts,sbox(pts),0.4,100,100)
> kl = ilayer(k,name="density",colorRamp(c("blue","red")))
> pts = data.frame(pts)
> coordinates(pts) <- cbind(pts[,1],pts[,2])
> ptsl = layer(pts,"Points",lstyle(fillColor="white",strokeColor="black"))
> osmMap(kl,ptsl)
```

map



These examples are fully explained in the `demo(osmMap)` which displays a PDF document.

2 Importing map tiles

Services like OpenStreetMap (http://www.openstreetmap.org/) produce map tiles for use in web map applications. These tiles can be downloaded as image

files. These functions grab tile images for use in R graphics by converting them to SpatialPixelDataFrame objects from the sp-package.

The main function is `getTiles`, which gets tiles given a spatial region, a zoom level, and a path to the tile server.

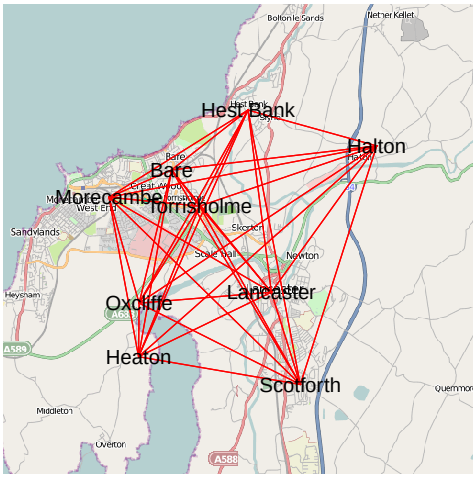
Always make sure you check the copyright and usage restrictions of any tile server you use. By default this package will cache tiles in a temporary R-session folder so that tiles should only be downloaded once per R-session. These folders are deleted at the end of an R session so you might want to create a more persistent folder for keeping downloaded tiles.

The tile usage policy for OpenStreetMap's "Mapnik" tiles is here:
`http://wiki.openstreetmap.org/wiki/Tile_usage_policy`

This example gets tiles for Lancaster and plots it. It then uses my `geonames` package to get some nearby places and draw the connections between them:

```
library(geonames)
```

```
>  lancs = getTiles(c(-2.842,-2.7579),c(54.0295,54.063),
+           zoom = 12,path="http://tile.openstreetmap.org/")
>  image(lancs)
>  local = GNfindNearbyPlaceName(lat=54.0475,lng=-2.800312,radius=5)
>  np = dim(local)[1]
>  pairs = cbind(rep(1:np,np),rep(1:np,rep(np,np)))
>  segments(local$lng[pairs[,1]],local$lat[pairs[,1]],
+           local$lng[pairs[,2]],local$lat[pairs[,2]],col="red")
>  text(local$lng,local$lat,local$name)
```



More information on `getTiles` can be found in the help for that function - type `help(getTiles)` for that.