Peer_Review_Cholera

Marc

15/04/2020

Instructions

1. From numerical data, draw map in John Snow's spirit. Show death places with markers whose size indicates number of deaths. Show water pumps on the same map with a different symbol.
2. Try to find different ways to show that the Broad street pump is at the center of the outbreak. (Density of deaths in the neighbourhood ? Other approaches ?).
3. Submit work on FUN.

Use `ggmap` with OpenStreetMaps as map template with `source="osm"`.

Importing data

There is already an version of John Snow's data sets and map in the package `HistData` in `Snow.deaths` and `Snow.pumps`. However, the coordinates fit John Snow's map and no other maps.

First download and extract the zip file from data sets available at this adress : http://rtwilson.com/downloads/SnowGIS_SHP.zip. It is the first link on Robin's blog.
Two files of interest are the `Cholera_Deaths.shp` and the `Pumps.shp` files containing death people and water pumps coordinates.
The next code lines test if the file `Cholera_Deaths.shp`exists. If not, you have to download it manually as I do not know how to unzip documents in R . . .

```
if(file.exists("SnowGIS_SHP/Cholera_Deaths.shp") == FALSE){
  print("You must download the zip file as explained above")
} else{
  print("Files already downloaded")
}
```

```
## [1] "Files already downloaded"
```

Reading the data

In order to read `.shp` files we need the `maptools` package.
The next code lines test if maptools is installed. If not, it does and load it :

```
if(!require("maptools")){
  install.packages("maptools")
  library("maptools")
}
```

Therefore, we can read the data with the `readShapePoints` function :

```
Deaths <- readShapePoints("SnowGIS_SHP/Cholera_Deaths")
```

```
## Warning: readShapePoints is deprecated; use rgdal::readOGR or sf::st_read
```

```
head(Deaths)
```

```
##            coordinates Id Count
## 0 (529308.7, 181031.4)  0     3
## 1 (529312.2, 181025.2)  0     2
## 2 (529314.4, 181020.3)  0     1
## 3 (529317.4, 181014.3)  0     1
## 4 (529320.7, 181007.9)  0     4
## 5   (529336.7, 181006)  0     2
```

We have coordinates, Id (I do not know what it is but I think it is not important) and Count that are the number of deaths at this location.
To extract coordinates, we can use the `coords`function.

```
head(Deaths@coords)
```

```
##   coords.x1 coords.x2
## 0  529308.7  181031.4
## 1  529312.2  181025.2
## 2  529314.4  181020.3
## 3  529317.4  181014.3
## 4  529320.7  181007.9
## 5  529336.7  181006.0
```

The number of recensed deaths is :

```
sum(Deaths$Count)
```

```
## [1] 489
```

It seems that this is less than recorded deaths (616) displayed on the FUN website but let's suppose that it Robin's fault…

We can also read the file with the Pumps coordinates :

```
Pumps <- readShapePoints("SnowGIS_SHP/Pumps")
```

```
## Warning: readShapePoints is deprecated; use rgdal::readOGR or sf::st_read
```

```
head(Pumps)
```

```
##            coordinates Id
## 0 (529396.5, 181025.1)  0
## 1 (529192.5, 181079.4)  0
## 2 (529183.7, 181193.7)  0
## 3 (529748.9, 180924.2)  0
## 4 (529613.2, 180896.8)  0
## 5 (529453.6, 180826.4)  0
```

There are 6 pumps.

Extracting london map

For this purpose we will need the `ggmap` package.
The next code lines test if ggmap is installed. If not, it does and load it :

```
if(!require("ggmap")){
  install.packages("ggmap")
  library("ggmap")
}
```

I got the london coordinates centrered on Broad Street on Stamen Maps website :

```
london <- c(left = -0.14454, bottom = 51.51139, right = -0.13119, top = 51.51630)
```

We will ge the map based on the above coordinates with streets names with the `maptype = "toner"`.

```
london_map = get_stamenmap(london, zoom = 17, maptype = "toner")
```

```
## Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL.
```

We can display the map with ggmap.

```
map <- ggmap(london_map)
```

Displaying the Cholera deaths and pumps on the map

We will first extract the points and their coordinates from the `.shp` file.
Remember that we put the Deaths points in the `Deaths` variable and the Pumps points in the `Pumps` variable.

```
Deaths_coord <- data.frame(Deaths@coords)
Pumps_coord <- data.frame(Pumps@coords)
```

Unfortunetely, both coordinates are in *OSGB36 National Grid* reference while our map is in classic *decimal degrees* reference.
Fortunetely, I found on the internet this following code that works but I do not know exactly how it works …
It seems that it specifies the *Coordinate Reference System (CRS)* first to the Deaths and Pumps coordinates with the `CRS` function.
And then, it transforms the system to a more classic longitude and lattitude system with the `sptransform` function.

```
coordinates(Deaths_coord)=~coords.x1+coords.x2
coordinates(Pumps_coord)=~coords.x1+coords.x2
proj4string(Deaths_coord)=CRS("+init=epsg:27700")
proj4string(Pumps_coord)=CRS("+init=epsg:27700")
Deaths_coord = spTransform(Deaths_coord,CRS("+proj=longlat +datum=WGS84"))
Pumps_coord = spTransform(Pumps_coord,CRS("+proj=longlat +datum=WGS84"))
df_Deaths=data.frame(Deaths_coord@coords)
df_Pumps=data.frame(Pumps_coord@coords)
```

Let’s see the results :

```
head(df_Deaths)
```

```
##    coords.x1 coords.x2
## 0 -0.1379301  51.51342
## 1 -0.1378831  51.51336
## 2 -0.1378529  51.51332
## 3 -0.1378120  51.51326
## 4 -0.1377668  51.51320
## 5 -0.1375369  51.51318
```

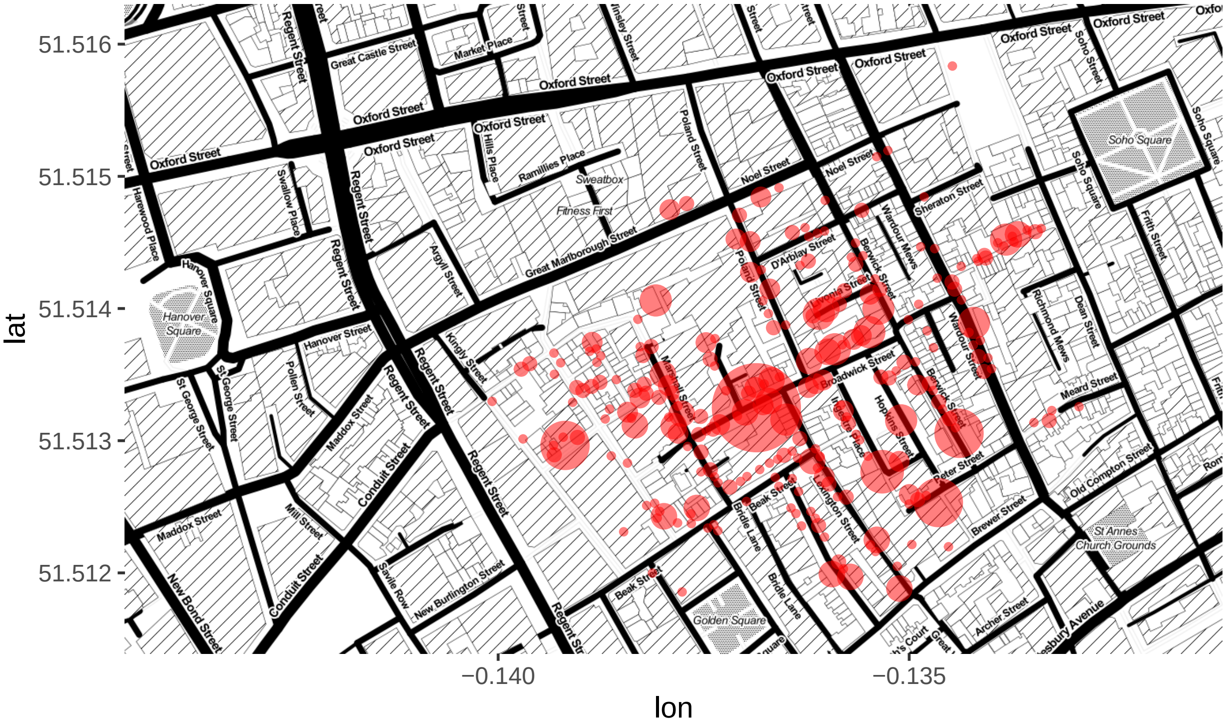
```r
head(df_Pumps)
```

```
##    coords.x1 coords.x2
## 0 -0.1366679  51.51334
## 1 -0.1395862  51.51388
## 2 -0.1396710  51.51491
## 3 -0.1316299  51.51235
## 4 -0.1335944  51.51214
## 5 -0.1359191  51.51154
```

Looks fine for both points :)!

Let's project the Deaths points on our map with the `geom_point`function in red and with a size depending on the number of deaths recorded at the particular coordinates :

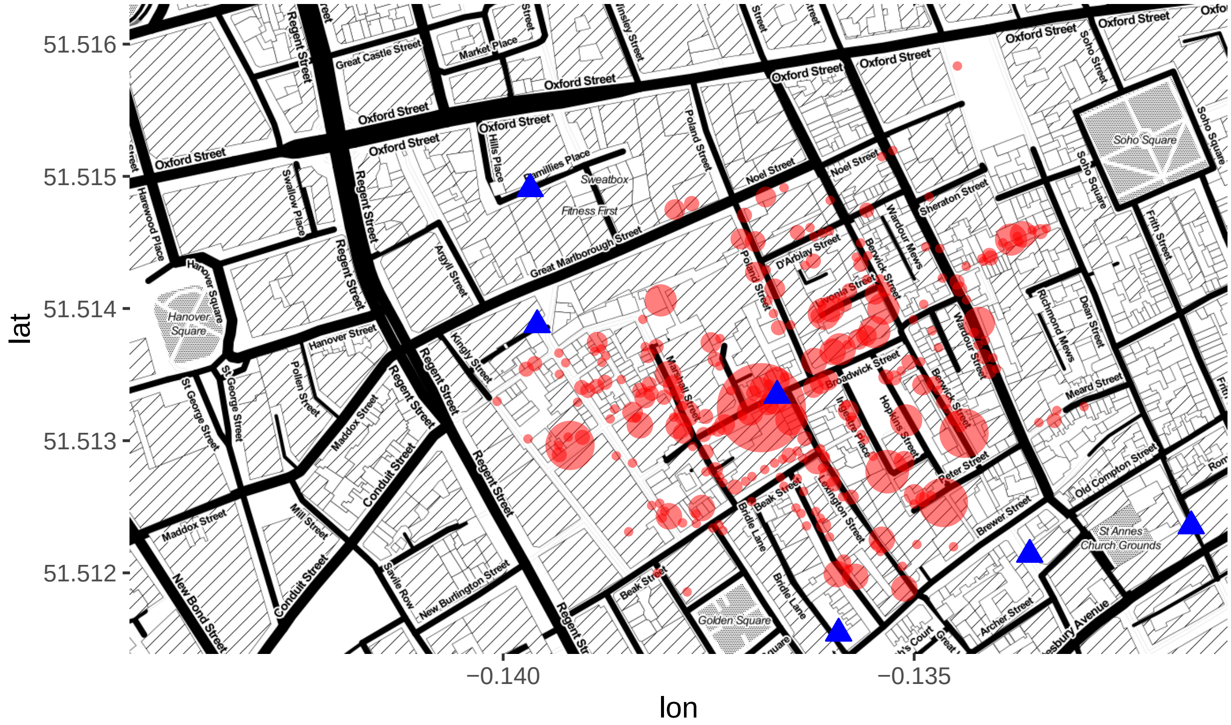
```r
map + geom_point(df_Deaths, mapping = aes(coords.x1, coords.x2), color = "red", size = Deaths$Count, alp
```



Let's add the pumps in another color and symbol :

```r
map + geom_point(df_Deaths, mapping = aes(coords.x1, coords.x2), color = "red", size = Deaths$Count, alp
  geom_point(df_Pumps, mapping = aes(coords.x1, coords.x2), color = "blue", shape = 24, size = 3, fill =
```

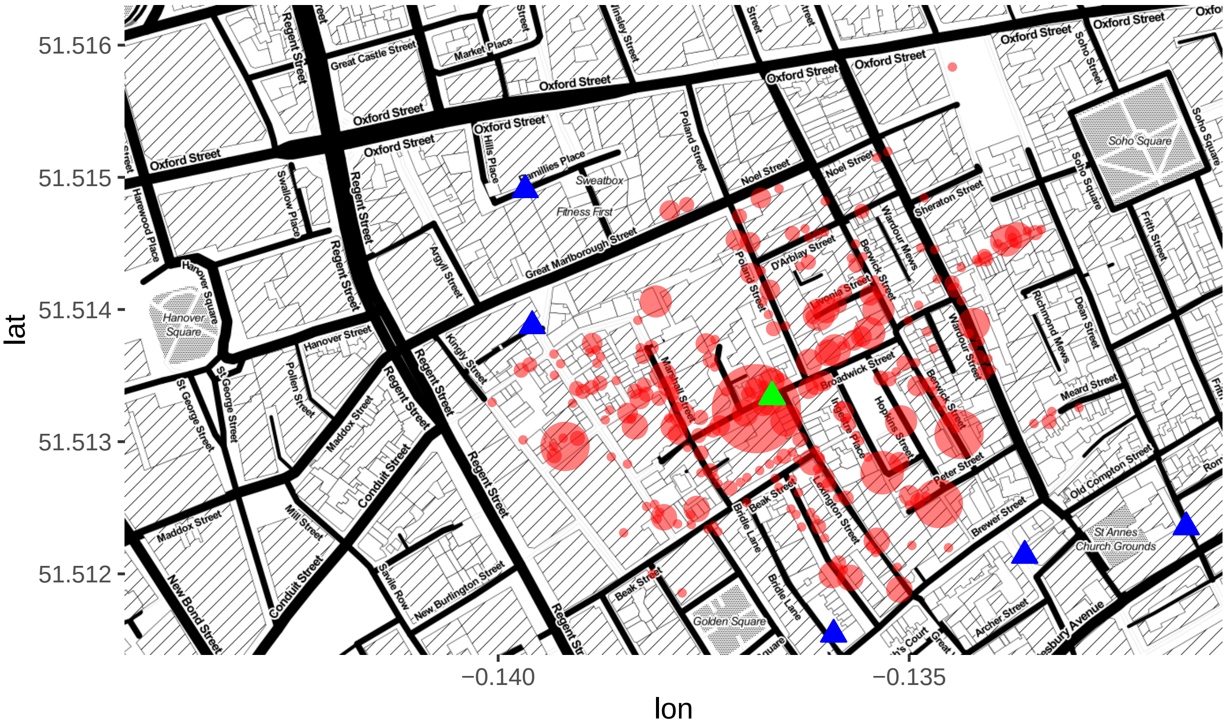
```
## Warning: Removed 2 rows containing missing values (geom_point).
```



Highlighting the Broad street pump (number 1 in the coordinates list :

```
map + geom_point(df_Deaths, mapping = aes(coords.x1, coords.x2), color = "red", size = Deaths$Count, alp
  geom_point(df_Pumps, mapping = aes(coords.x1, coords.x2), color = "blue", shape = 24, size = 3, fill
  geom_point(df_Pumps, mapping = aes(coords.x1[1], coords.x2[1]), color = "green", shape = 24, size = 3
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



Analysis to show that the Broad street pump is at the center of the outbreak