Reading Set 6

Ziji Zhou

Due by 10pm ET on Monday

Reading Set Information

A more thorough reading and light practice of the textbook reading prior to class allows us to jump into things more quickly in class and dive deeper into topics. As you actively read the textbook, you will work through the Reading Sets to help you engage with the new concepts and skills, often by replicating on your own the examples covered in the book.

These should be completed on your own without help from your peers. While most of our work in this class will be collaborative, it is important each individual completes the active readings. The problems should be straightforward based on the textbook readings, but if you have any questions, feel free to ask me!

GitHub Workflow

1. Before editing this file, verify you are working on the copy saved in *your* repo for the course (check the filepath and the project name in the top right corner).
2. Before editing this file, make an initial commit of the file to your repo to add your copy of the problem set.
3. Change your name at the top of the file and get started!
4. You should *save, knit, and commit* the .Rmd file each time you've finished a question, if not more often.
5. You should occasionally *push* the updated version of the .Rmd file back onto GitHub. When you are ready to push, you can click on the Git pane and then click **Push**. You can also do this after each commit in RStudio by clicking **Push** in the top right of the *Commit* pop-up window.
6. When you think you are done with the assignment, save the pdf as "*Name_thisfilename_date*.pdf" (it's okay to leave out the date if you don't need it) before committing and pushing (this is generally good practice but also helps me in those times where I need to download all student homework files).

Gradescope Upload

For each question (e.g., 3.1), allocate all pages associated with the specific question. If your work for a question runs onto a page that you did not select, you may not get credit for the work. If you do not allocate *any* pages when you upload your pdf, you may get a zero for the assignment.

You can resubmit your work as many times as you want before the deadline, so you should not wait until the last minute to submit some version of your work. Unexpected delays/crises that occur on the day the assignment is due do not warrant extensions (please submit whatever you have done to receive partial credit).

Problem 1 **Spatial data structures** Section 17.1 introduces *shapefiles*, and includes an example of working with a shapefile to re-create Snow's cholera map.

1.1 Load the **sf** package in the `setup` code chunk. Verify your working directory is the folder *this* file is in (see the README file of the labs folder). If it's not already created, make sure you have a *data* subfolder in this directory.

1.2 Run the code below line-by-line to understand what each part is doing (reminder: you can use *command + enter* or *ctrl + enter* to run one selected or highlighted set of code at a time). Confirm that you get a figure similar to that of Figure 17.3 in the textbook.

```
# Download SnowGIS_SHP zip file
download.file("http://rtwilson.com/downloads/SnowGIS_SHP.zip",
              destfile = "data/SnowGIS_SHP.zip")

# Unzip file in same folder
unzip(zipfile = "data/SnowGIS_SHP.zip",
      exdir = "data")
```

```
# Create filepath to unzipped files so we don't need to re-type
data_path <- "data/SnowGIS_SHP"

# List files in SnowGIS_SHP
list.files(data_path)
```

```
 [1] "Cholera_Deaths.dbf"          "Cholera_Deaths.prj"
 [3] "Cholera_Deaths.sbn"          "Cholera_Deaths.sbx"
 [5] "Cholera_Deaths.shp"          "Cholera_Deaths.shx"
 [7] "OSMap_Grayscale.tfw"         "OSMap_Grayscale.tif"
 [9] "OSMap_Grayscale.tif.aux.xml" "OSMap_Grayscale.tif.ovr"
[11] "OSMap.tfw"                   "OSMap.tif"
[13] "Pumps.dbf"                   "Pumps.prj"
[15] "Pumps.sbx"                   "Pumps.shp"
[17] "Pumps.shx"                   "README.txt"
[19] "SnowMap.tfw"                 "SnowMap.tif"
[21] "SnowMap.tif.aux.xml"         "SnowMap.tif.ovr"
```

```
# List layers
st_layers(data_path)
```

```
Driver: ESRI Shapefile
Available layers:
      layer_name geometry_type features fields
1 Cholera_Deaths         Point      250      2
2          Pumps         Point        8      1
```

```
# Load second layer
cholera_deaths <- st_read(data_path, layer = "Cholera_Deaths")
```

```
Reading layer 'Cholera_Deaths' from data source
  '/Users/zijizhou/Documents/Amherst/STAT 231 DATA SCIENCE/datascience/problem_sets/rs6/data/SnowG
  using driver 'ESRI Shapefile'
Simple feature collection with 250 features and 2 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 529160.3 ymin: 180857.9 xmax: 529655.9 ymax: 181306.2
Projected CRS: OSGB 1936 / British National Grid
```

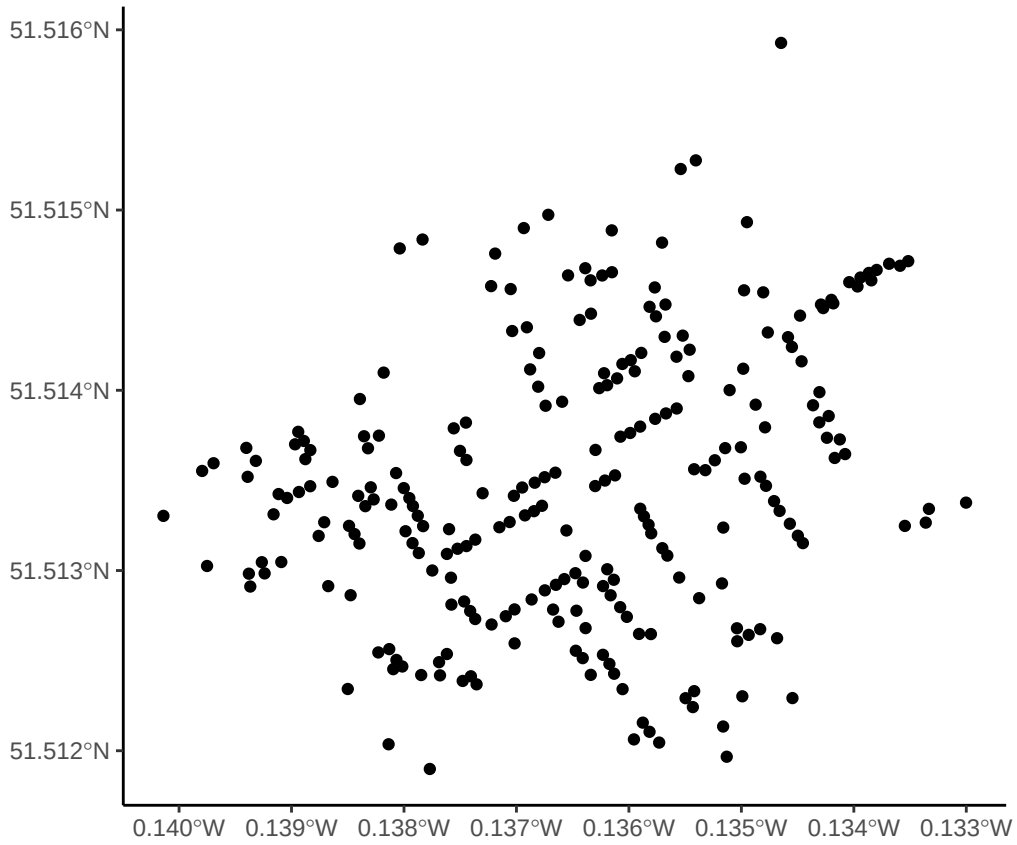
```
class(cholera_deaths)
```

```
[1] "sf"         "data.frame"
```

```
head(cholera_deaths)
```

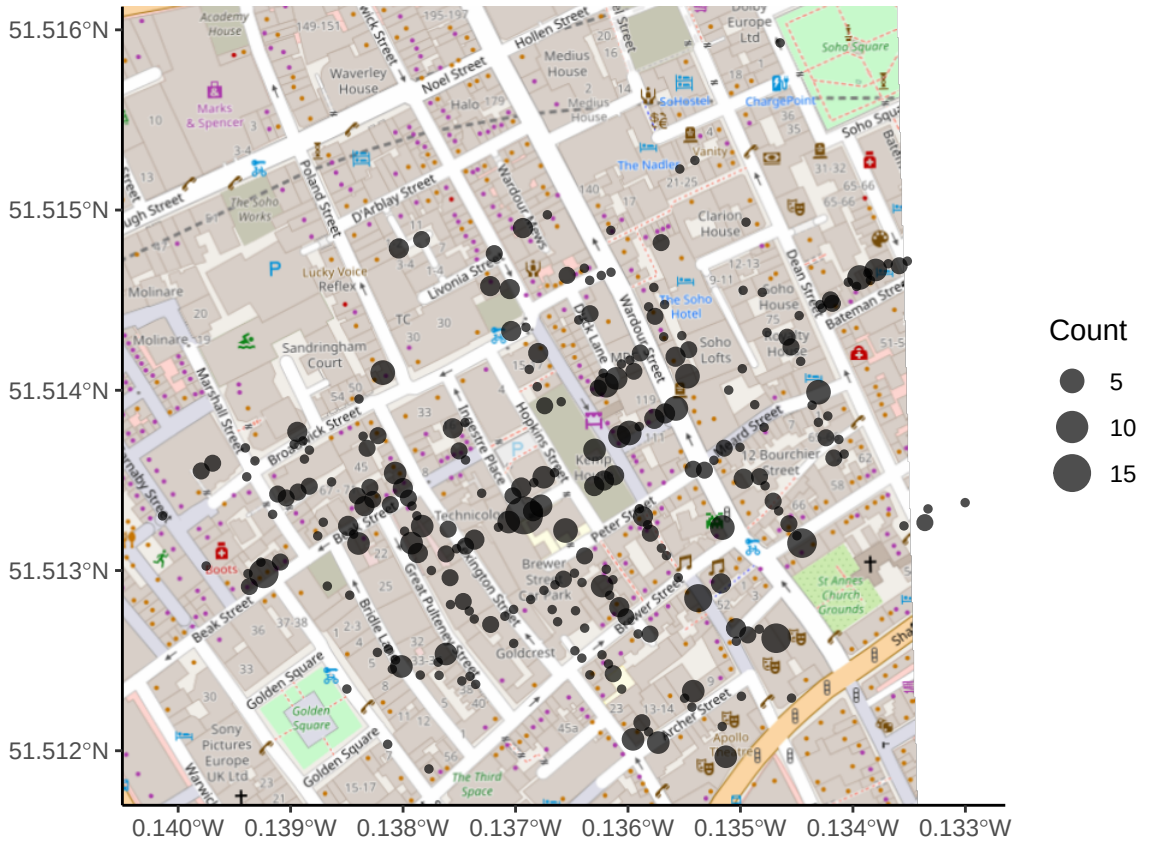
```
Simple feature collection with 6 features and 2 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 529308.7 ymin: 181006 xmax: 529336.7 ymax: 181031.4
Projected CRS: OSGB 1936 / British National Grid
  Id Count                  geometry
1  0     3 POINT (529308.7 181031.4)
2  0     2 POINT (529312.2 181025.2)
3  0     1 POINT (529314.4 181020.3)
4  0     1 POINT (529317.4 181014.3)
5  0     4 POINT (529320.7 181007.9)
6  0     2   POINT (529336.7 181006)
```

```
# Context-less plot
ggplot(cholera_deaths) +
  geom_sf()
```



1.3 Now use the **ggspatial** package to overlay the London street map. Make sure you (install then) load the **ggspatial** package in the `setup` code chunk before running the code. What is wrong with this map?

```
ggplot(cholera_deaths) +
  annotation_map_tile(type = "osm", zoomin = 0) +
  geom_sf(aes(size = Count), alpha = 0.7)
```

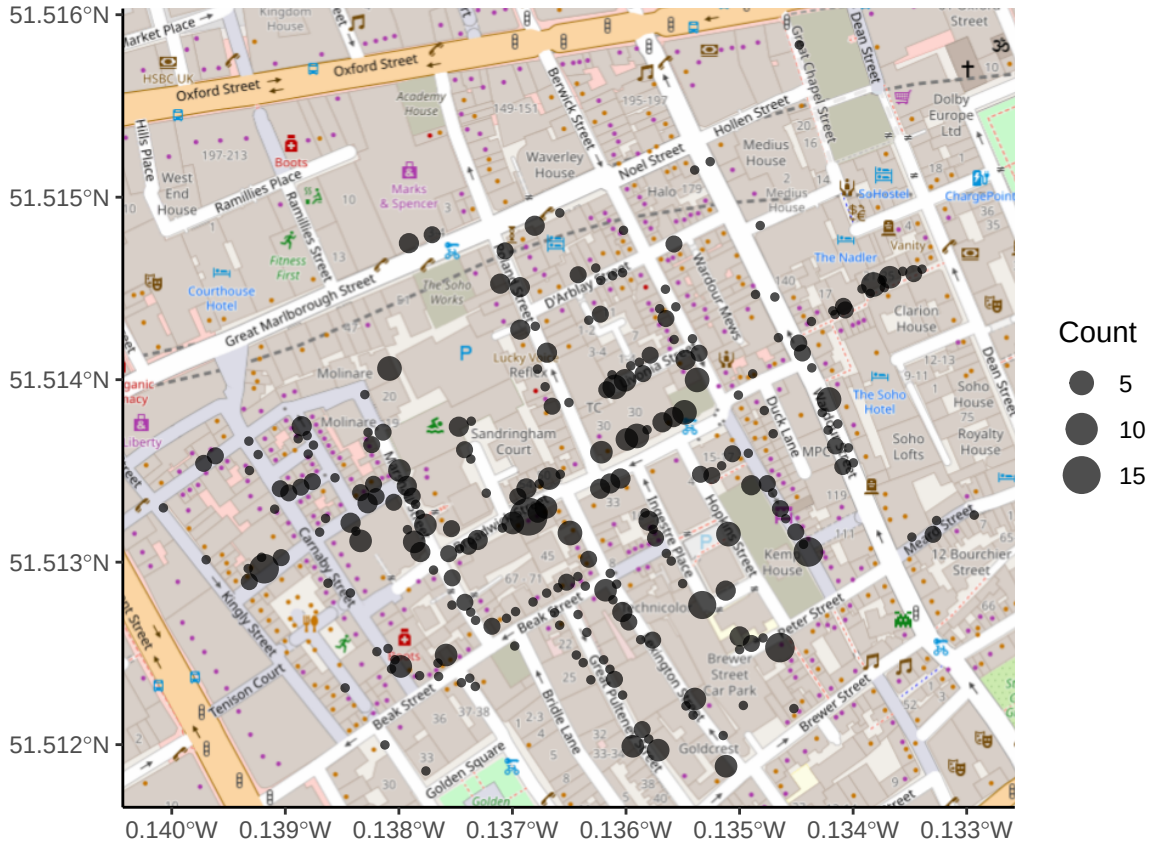


The points are not matching up with the map projection. This is because of the different formats of map coordinate data stored on the projection and data points.

1.4 Set the coordinates from the cholera data as the `espg:27700` coordinate system using `st_set_crs()`, then transform them to the `espg:4326` system using `st_transform()`, and finally plot the new, correctly projected data.

```
cholera_latlong <- cholera_deaths %>%
  st_set_crs(27700) %>%
  st_transform(4326)

ggplot(cholera_latlong) +
  annotation_map_tile(type = "osm", zoomin = 0) +
  geom_sf(aes(size = Count), alpha = 0.7)
```

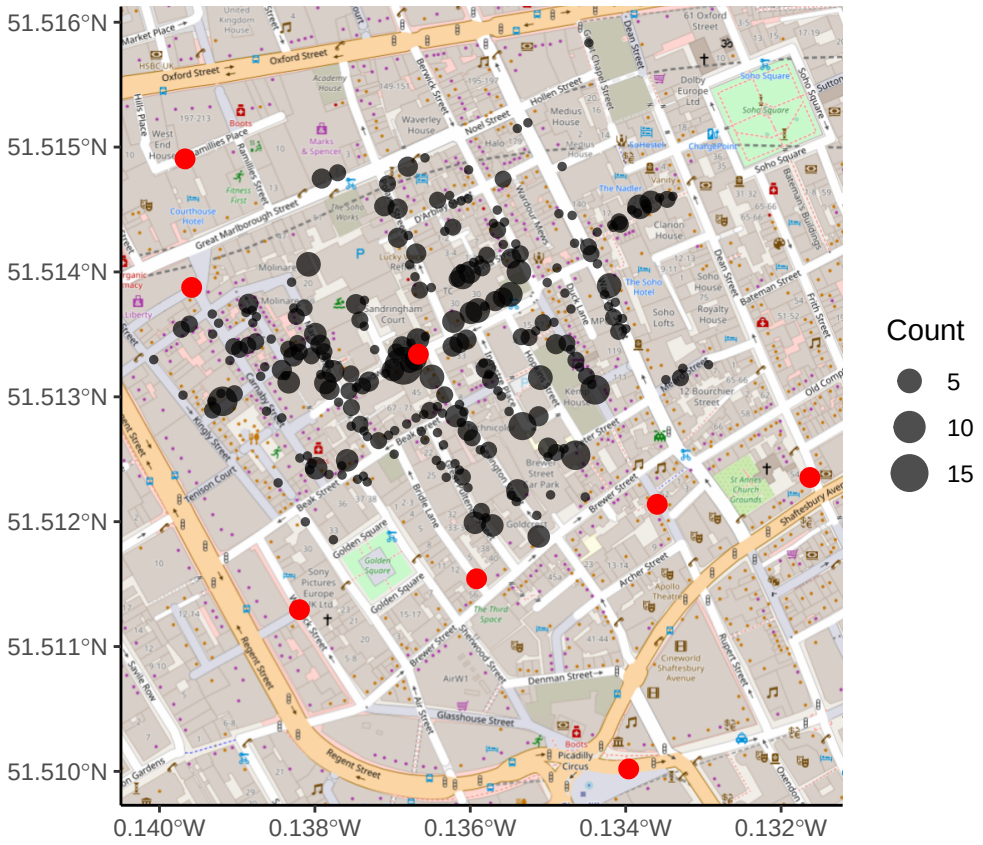


1.5 Repeat the layer loading and coordinate transformation procedure to add the water pumps to the plot.

```
pumps_latlong <- st_read(data_path, layer = "Pumps") %>%
  st_set_crs(27700) %>%
  st_transform(4326)
```

```
Reading layer 'Pumps' from data source
  '/Users/zijizhou/Documents/Amherst/STAT 231 DATA SCIENCE/datascience/problem_sets/rs6/data/SnowG
  using driver 'ESRI Shapefile'
Simple feature collection with 8 features and 1 field
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 529183.7 ymin: 180660.5 xmax: 529748.9 ymax: 181193.7
Projected CRS: OSGB 1936 / British National Grid
```

```
ggplot(cholera_latlong) +
  annotation_map_tile(type = "osm", zoomin = 0) +
  geom_sf(aes(size = Count), alpha = 0.7) +
  geom_sf(data = pumps_latlong, size = 3, color = "red")
```



1.6 Finally, try out the code below to create a dynamic map using the **leaflet** package. Zoom in and out of the map to confirm that (1) there is a death in the middle of Hopkins Street, and (2) there is a pump near the intersection of Broadwick Street and Lexington Street.

```
# create dynamic map
leaflet() %>%
  addTiles() %>%
  addCircleMarkers(data = cholera_latlong,
                   radius =  ~ Count,
                   color = "navy",
                   stroke = FALSE,
                   fillOpacity = 0.7) %>%
  addCircleMarkers(data = pumps_latlong,
                   radius = 6,
                   color = "red",
                   stroke = FALSE,
                   fillOpacity = 0.7)
```

There is indeed a death right in the middle of Hopkins St as well as the pump on the intersection.