# Project 1

1] **Spatial Data Manipulation**

```
shape <- readOGR(dsn = "D:/UPitt/Studies/Sem 2/Spatial
DA/pgh_streets/pgh_streets/pgh_streets", layer = "pgh_streets")
```

**a.**

```
nrow(shape@data)
# The total number of road segments is 22222
```

**b.**

```
min(shape@data[["LENGTH"]])
# Minimum segment length is 3e-04.

max(shape@data[["LENGTH"]])
# Maximum segment length is 1.46654.

mean = mean(shape@data[["LENGTH"]])
print(mean)
# Mean segment length is 0.05979852.
```

**c.**

```
shape.filtered = shape[as.numeric(shape@data$LENGTH) >= mean, ]
cat("Map after filtering out the segments that are below the mean length is: ")
plot(shape.filtered)
```

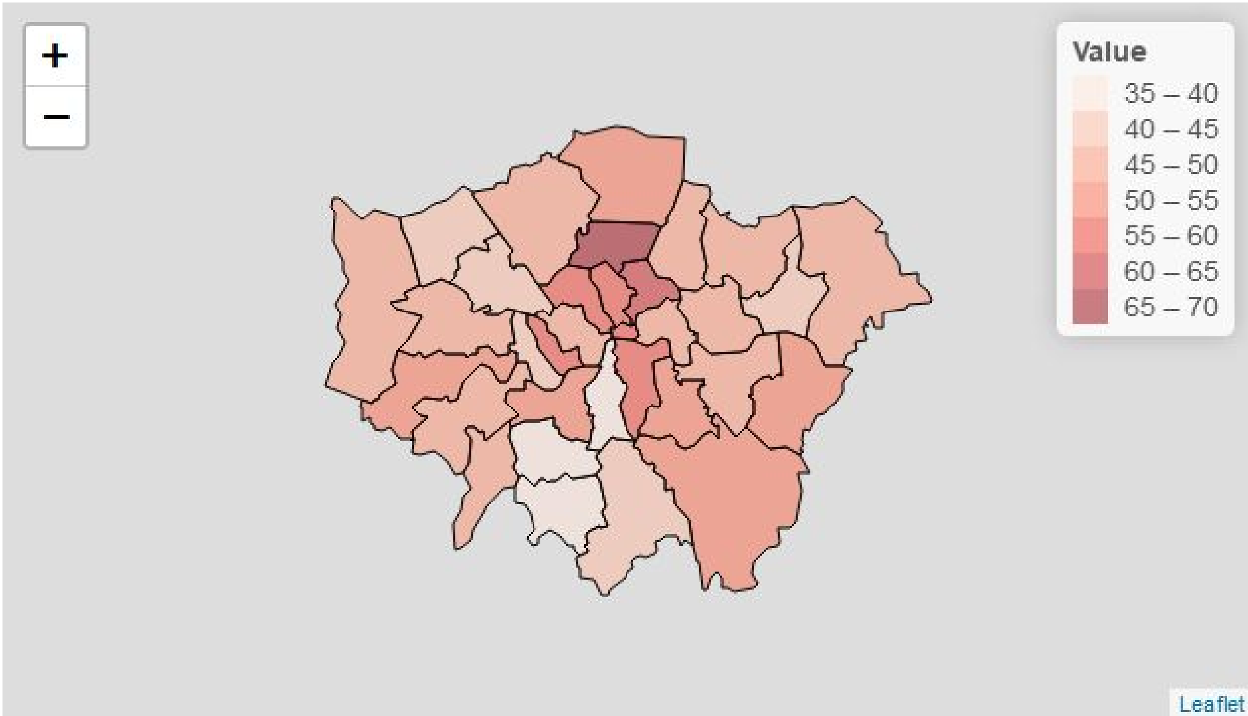
2] **Spatial Data Aggregation**

```
lnd <- get(load('D:/UPitt/Studies/Sem 2/Spatial DA/Assignments/Assignment 1/lnd.Rdata'))

stations <- get(load('D:/UPitt/Studies/Sem 2/Spatial DA/Assignments/Assignment 1/stations.Rdata'))
```

**a.**

```
  stations.m <- aggregate(stations[c("Value")], by = lnd, FUN = mean)
# Reproject coordinates
  stations_agg <- spTransform(stations.m, CRS("+init=epsg:4326"))
library(leaflet)
  qpal <- colorBin("Reds", stations_agg$Value, bins=5)
  leaflet(stations_agg) %>%
  addPolygons(stroke = TRUE,opacity = 1,fillOpacity = 0.5, smoothFactor = 0.5,
            color="black",fillColor = ~qpal(Value),weight = 1) %>%
   addLegend(values=~Value,pal=qpal,title="Value")
```




**Report**:
After observing the aggregation results, it is observed that the borough Lambeth in London has the least Value(mean value), and Haringey borough has the maximum Value(mean value). Mean value of the borough values is 49.11457. There are in total 14 borough with values greater than that of this mean value and 19 with values lesser than that of the mean value.

**b.**

```
stations@data["MidPoint"] =
(stations@data["coords.x1"]+stations@data["coords.x2"])/2
# R-squared value before aggregation
cat("R-squared value before aggregation is: ", summary(lm(Value~MidPoint,
stations))$adj.r.squared)
require(rgdal)
require(PBSmapping)
require(maptools)
stations.ps <- SpatialPolygons2PolySet(stations.m)
stations.ps.centroids <- calcCentroid(stations.ps, rollup=1)
```

```
stations.ps.centroids["MidPoint"] =
(stations.ps.centroids["X"]+stations.ps.centroids["Y"])/2
stations.ps.centroids["MeanValue"] = stations.m@data["Value"]
# R-squared value after aggregation
cat("R-squared value after aggregation is: ", summary(lm(MeanValue~MidPoint,
stations.ps.centroids))$adj.r.squared)
```

**Report**: After observing the regression results, it is observed that the R-squared value after aggregation(0.03866322) is higher than the R-squared value before aggregation(0.003260015). Before aggregation, the independent variable(x) in regression is the mid-point of the x and y coordinates of the railway stations. And, dependent variable(y) in regression is the value in each railway station.

After aggregation, we get the mean value in each polygon(borough) by taking the mean of the values of railway stations in that borough. This is the dependent variable(y) in regression. And, we get the coordinates of the centroid of each polygon(burough) using calcCentroid function. Independent variable(x) in regression is the value of the mid-point of the x and y coordinates of the centroid of each polygon(burough).

Aggregation of observations leads to an outcome that is closer to the mean of the overall data, so that after aggregation, the new data are likely to be more tightly clustered around a regression line and thus to have a higher R-squared value. Thus, aggregation schemes are more likely to producing better fits than the original disaggregated data.