Problem Set 4

Shisham Adhikari

Math 345

Due: Before class on Thursday, 03/04/2021

Collaborators: Aditya, Grisha

Goals of this lab

1. Map a (possibly) new-to-you point dataset.
2. Practice with density and distance-based point pattern analyses. Load any packages you need:

```
pacman::p_load(tidyverse, spdep, ggmap, sf, spatstat, maptools, raster)
```

Problem 1

Get some energy generation data from the US Energy Atlas. a) Check out the power plants datasets available from the EIA's US Energy Atlas. Download the data for at least 2 energy sources. You can use whichever format you prefer.

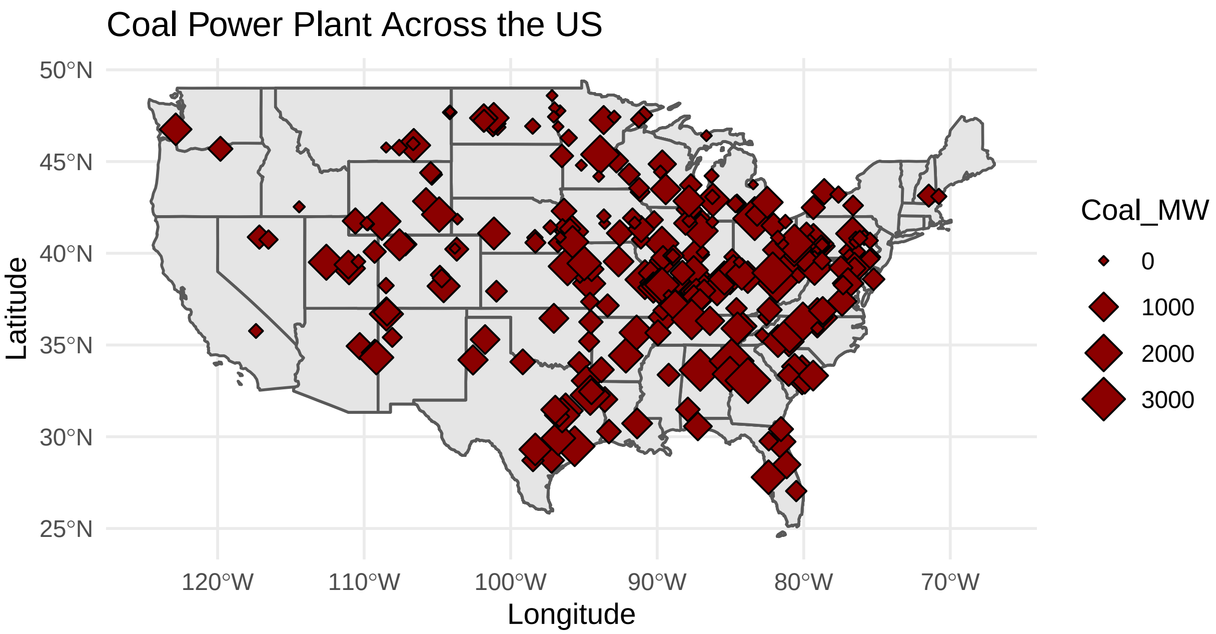
```
coal <- read_csv("coal.csv")
```

Since coal has more than 100 observations, just choosing one energy source is enough.

b) Graph your data. Up to you whether you want to cover the whole area, or pick a subset of the US. Just make sure you have at least 100 points.

```
us_states <- spData::us_states
coal_with_geom <- left_join(us_states, coal, by = c(NAME = "State"))
coal_with_geom_sp <- as(coal_with_geom, "Spatial")
coal_sf <- st_as_sf(coal, coords = c("Longitude", "Latitude"),
                    crs = ’+proj=longlat +datum=WGS84’)

#static map
ggplot(data = coal_with_geom) +
    geom_sf() +
    geom_point(aes(x = Longitude, y = Latitude, size=Coal_MW),
        shape = 23, fill = "darkred") +
  labs(title="Coal Power Plant Across the US")+
  theme_minimal()
```



Problem 2

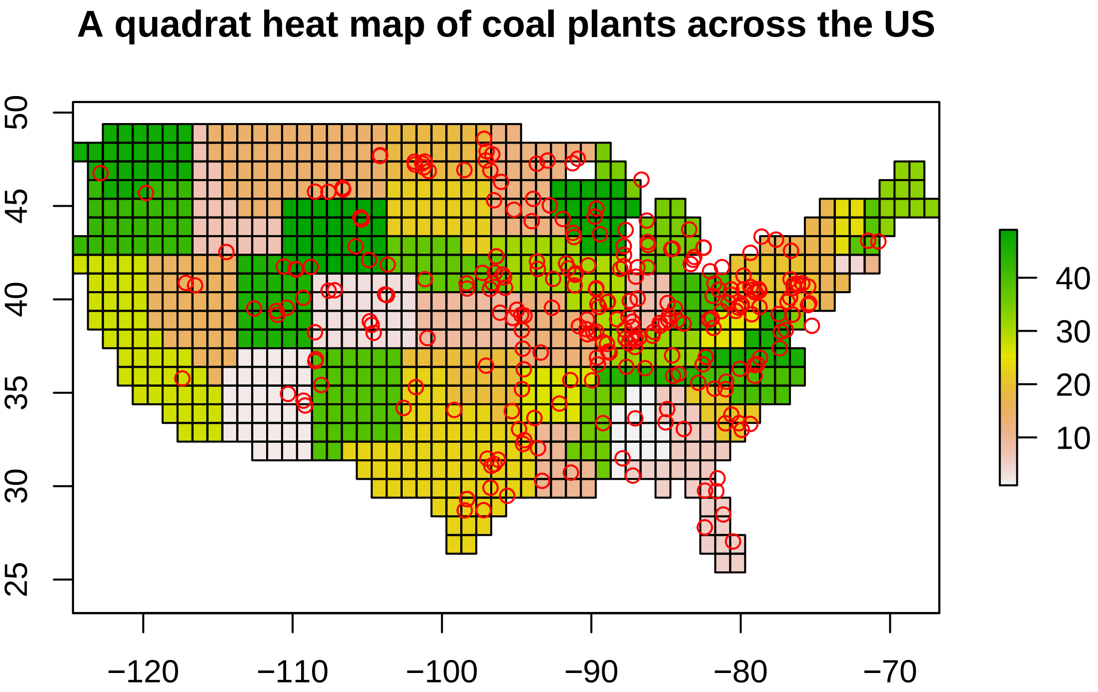
Let's play with the concepts of density. a) First, calculate the overall power plant density for your study area.

```
#converting sf to sp
us_sp <- as(us_states, 'Spatial')
coal_sp <- as(coal_sf, 'Spatial')
#Finding area for US
USArea <- raster::area(us_sp)
#Calculating power plan density for our study area
xy <- coordinates(coal_sp)
dens <- nrow(xy) / USArea
head(dens)
```

```
## [1] 2.161406e-09 9.787279e-10 1.072066e-09 2.227088e-08 1.913248e-09
## [6] 1.892287e-09
```

b) Now, create a grid over your study area with a resolution that makes sense to you. Create a quadrat heat map.

```
r <- raster(us_sp)
res(r) <- 1
g <- rasterize(us_sp, r)
plot(g, main = "A quadrat heat map of coal plants across the US")
quads <- as(g, 'SpatialPolygons')
plot(quads, add=TRUE)
points(coal, col='red')
```



Here, we see more concentration in the mid-west region.

c) Consider the points created by the intersection of your grid lines. Without invoking a density function, at each point, calculate the kernel density. Use a uniform kernel with several different bandwidth choices. Visualize your results.

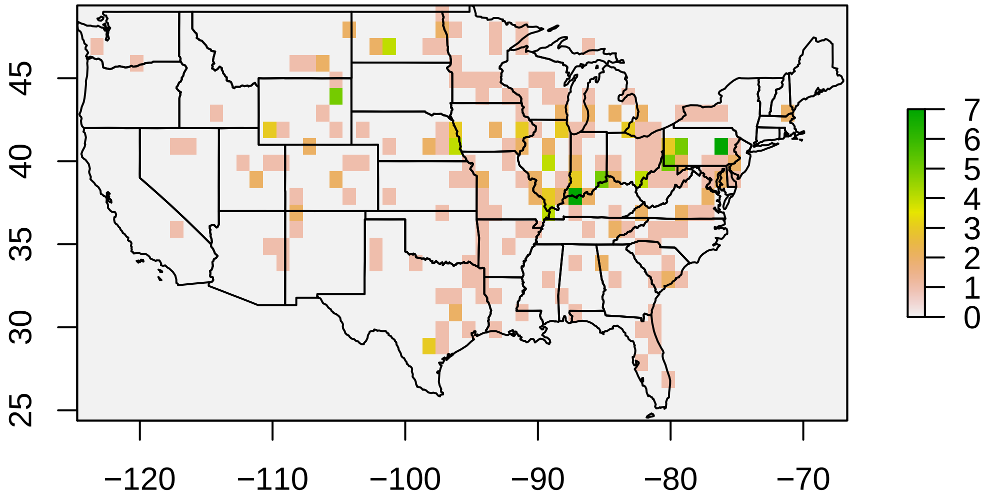
Methodology described: We use simple, or naive, density estimation process to calculate the kernel density without invoking a density function at each point. To do so, we first create a disk around each intersection of grid lines and count number of points inside the disk. To do so, we can calculate distance of every points from a kernel center and use a similar methodology as G function to check how many points are in the disk. We then calculate the area of each disk and area of each power plan to get density at each point. The process is very similar to the G-function, the key difference is we are here calculating density instead of neighrest neighbor. I could not fully exceute this methodology, so I implemented a simpler method.

The number of events in each quadrat can be counted using the 'rasterize' function. That function can be used to summarize the number of points within each cell and that way we can find the naive density function.

Kernel function K(x; r) for a uniform kernel is $\frac{1}{2}1_{([x]\leq 1)}$.

```
#Simple density estimation using counts on each quads
cl <- rasterize(coordinates(coal_sp), g, fun='count', background=0)
plot(cl, main="Kernel density based on counts inside each quadrat")
plot(us_sp, add=TRUE)
```

Kernel density based on counts inside each quadrat



Compare the results to the quadrat plot from part (b).

- We see that the plot in (c) is similar to (b) in that both of them detect high density of power plants in the mis-west region. The difference in, however, the plot in (b) is also showing high density in north west and south west region which is something we don't see explicitly in (c). It makes sense as the heat map across the entire state is likely to overestimate power plant density for that state.

d) Create a kernel density map using the density function and kernel function of your choice.

```
us_transform <- sf::st_transform(us_states, crs = 6345)
us_transform <- as(us_transform, "Spatial")
coal_transform <- sf::st_transform(coal_sf, crs = 6345)
coal_transform <- as(coal_transform, "Spatial")
usOwin <- as(us_transform, "owin")
class(usOwin)
```

```
## [1] "owin"
```

```
usOwin
```

```
## window: polygonal boundary
## enclosing rectangle: [-2717902.5, 2082513.2] x [2726487, 6084194] units
```

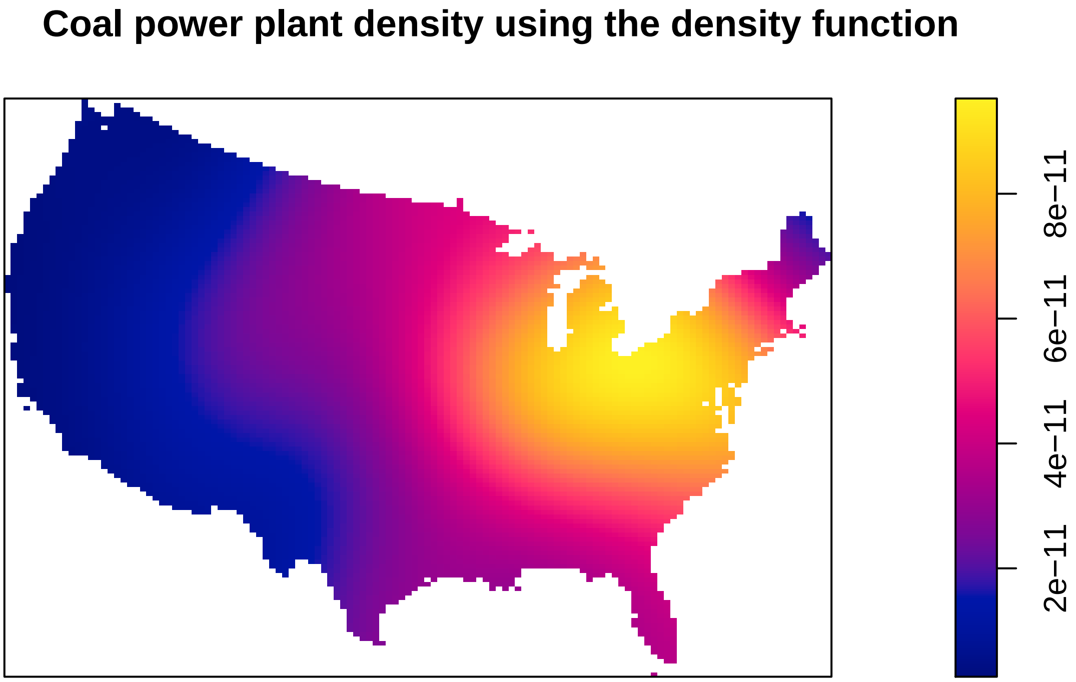
```
#Extract coordinates from SpatialPointsDataFrame:
#Now we can create a 'ppp' (point pattern) object
pts <- coordinates(coal_transform)
head(pts)
```

```
##      coords.x1 coords.x2
## [1,]  -120885.3   3562751
## [2,]   413498.8   3484048
## [3,]  -457377.7   4585096
## [4,]  -458332.2   4554212
## [5,] -1988956.9   8555602
## [6,] -1427746.9   4130307
```

```
p <- ppp(pts[,1], pts[,2], window=usOwin)
ds <- density(p)
class(ds)
```

```
## [1] "im"
```

```
par(mar = c(1, 1, 1, 1.1))
plot(ds, main='Coal power plant density using the density function')
```



Problem 3

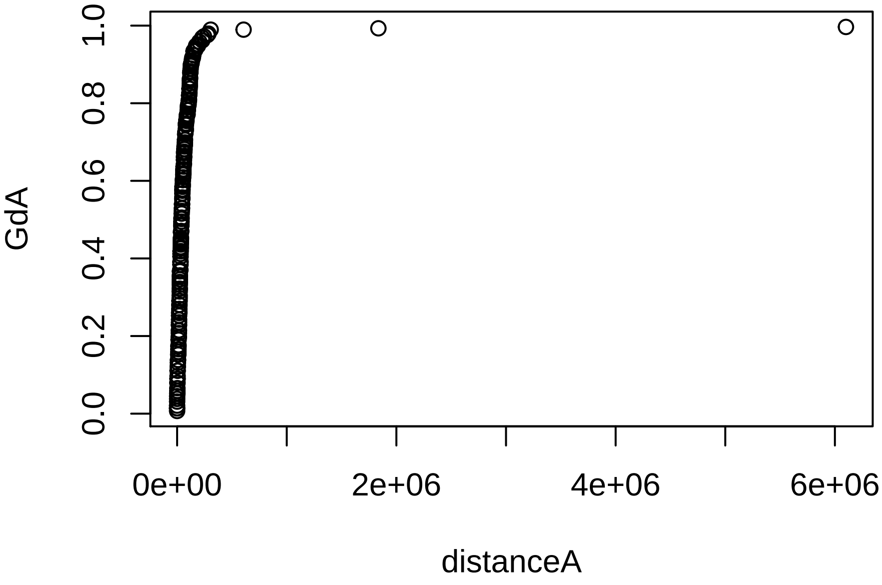
Let's do a few distance based calculations.

a) Calculate and plot the nearest neighbor G-function.

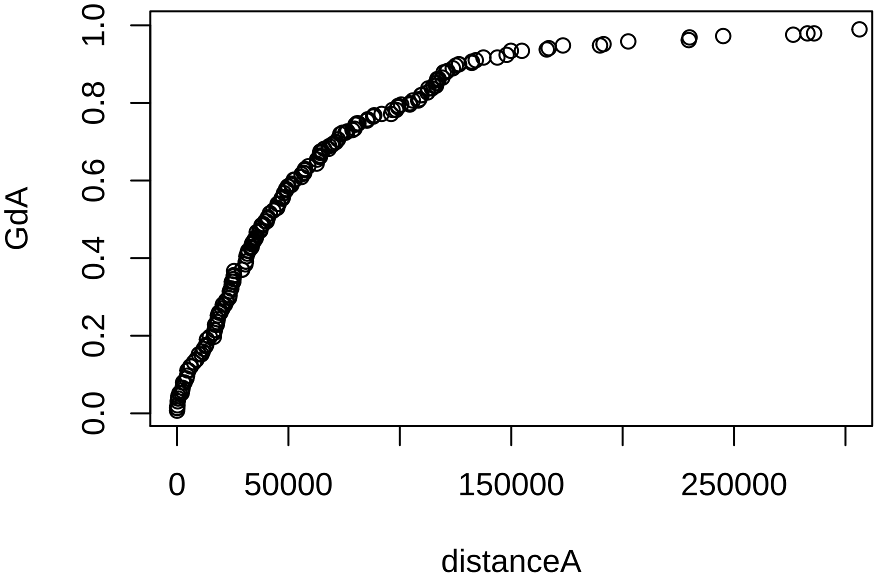
```
a <- dist(pts)
am <- as.matrix(a)
diag(am) <- NA
amin <- apply(am, 1, min, na.rm=TRUE)
# get the unique distances (for the x-axis)
max(amin)
```

```
## [1] 6100835
```

```
distanceA <- sort(unique(round(amin)))
# compute how many cases there with distances smaller that each x
GdA <- sapply(distanceA, function(x) sum(amin < x))
# normalize to get values between 0 and 1
GdA <- GdA / length(amin)
plot(distanceA, GdA)
```



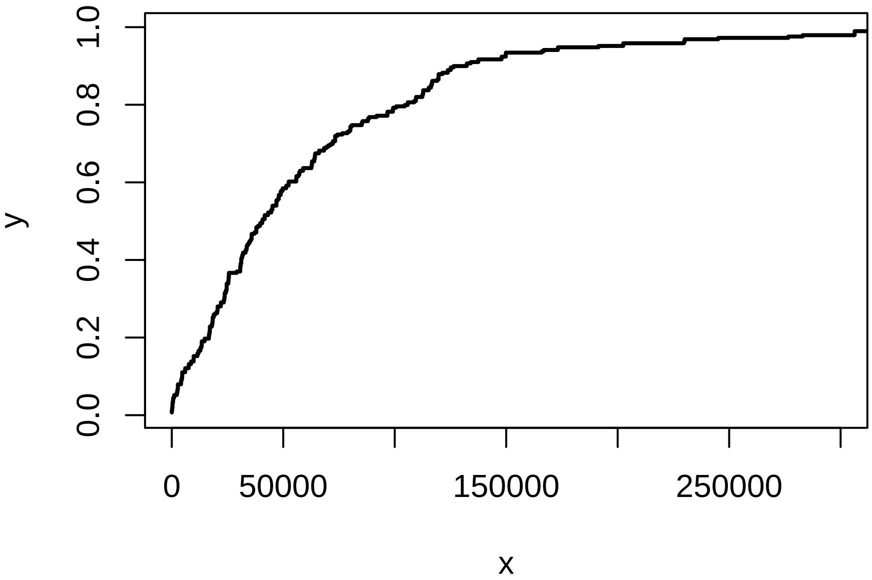
```r
# using xlim to exclude the extremes
plot(distanceA, GdA, xlim=c(0,300000))
```



```r
stepplot <- function(x, y, type='l', add=FALSE, ...) {
    x <- as.vector(t(cbind(x, c(x[-1], x[length(x)]))))
    y <- as.vector(t(cbind(y, y)))
  if (add) {
     lines(x,y, ...)
  } else {
       plot(x,y, type=type, ...)
  }
}
```

And use it for our G function data.

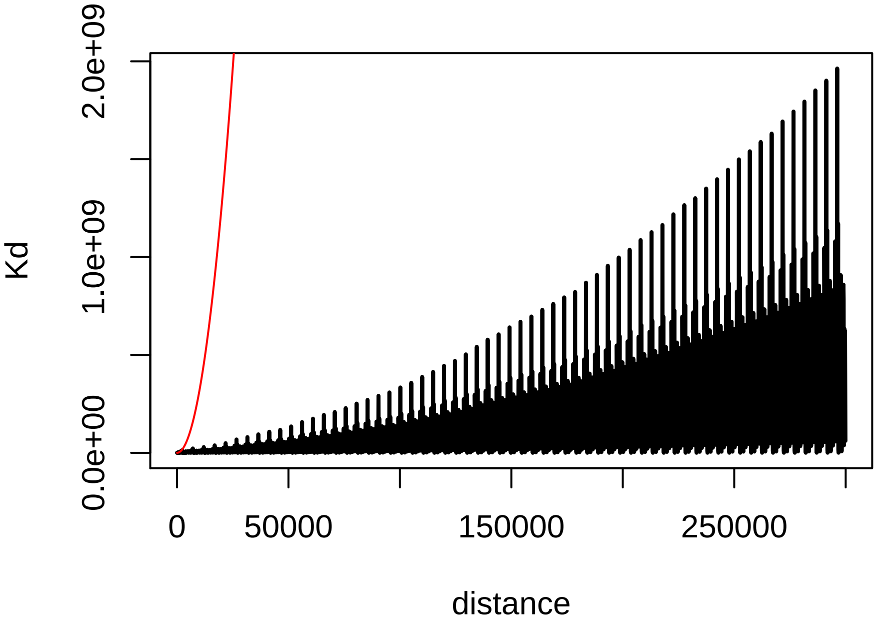
```
stepplot(distanceA, GdA, type='l', lwd=2, xlim=c(0,300000))
```



Interpretation: Based on the above G-function plot, 40% of points are within 50000units distance of a random point and above 80% of points are within 200000units distance of a random point.

b) [We'll learn about the K-function on Tuesday, so you'll probably want to wait to do this part until then.] Calculate and plot the K-function. Include the benchmark line for a poisson point process.

```
distance <- seq(1, 300000, 100)
Kd <- sapply(distance, function(x) sum(a < x)) # takes a while
Kd <- Kd / (length(Kd) * dens)
#The intensity (lambda) is the number of points divided by the number of quadrats.
plot(distance, Kd, type='l', lwd=2)
lines(distance, pi*(distance^2), type = "l", lty = 1, col="red")
```



```
#the benchmark line for a poisson point process for a K-function is K(t)=pi*t^2
```

In the above graph, k-function over the benchmark line represents the points are more clustered and below the line represents the points are less dispersed than a poisson point process. Based on the above graph, the points are more dispersed than a poisson point process.

c) Write up any conclusions you can make about the spatial distribution of power plants in your study area.

- Based on our plots, the spatial distribution of coal-based power plants across the US is positively correlated; there is higher density of plants in mid-west region than other region. This is seen more explicitly in the plot using density function. Based on the K-function analysis, however, we see that the power plants are more dispersed than a poisson point process which is interesting.