Project 3

Jiawen Qi (jiq10)

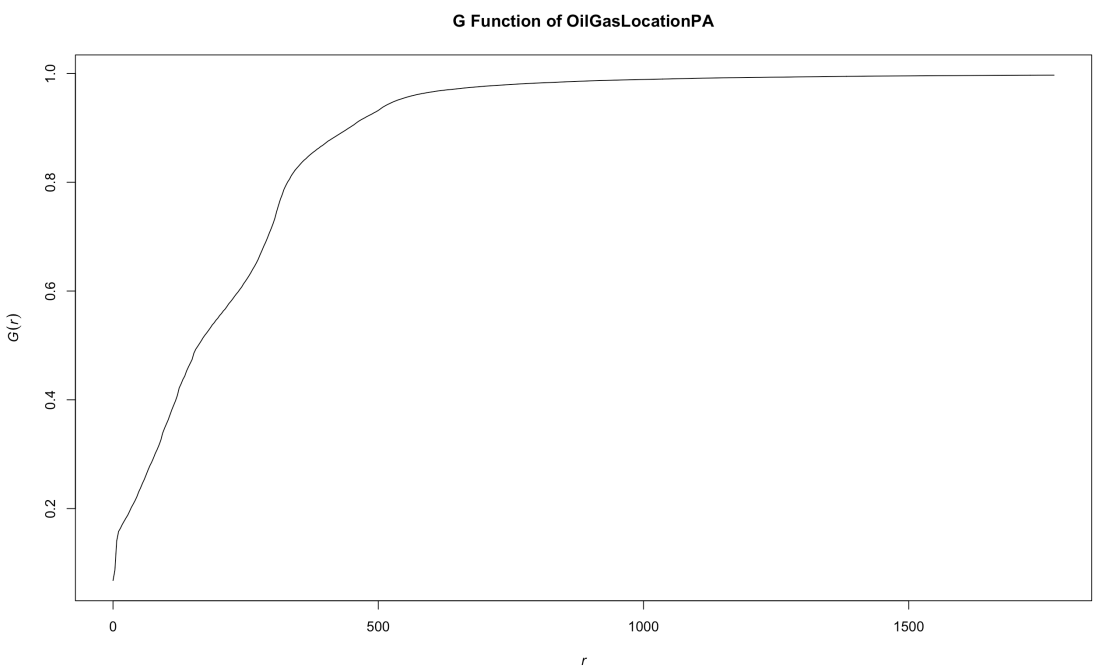
March 17, 2017

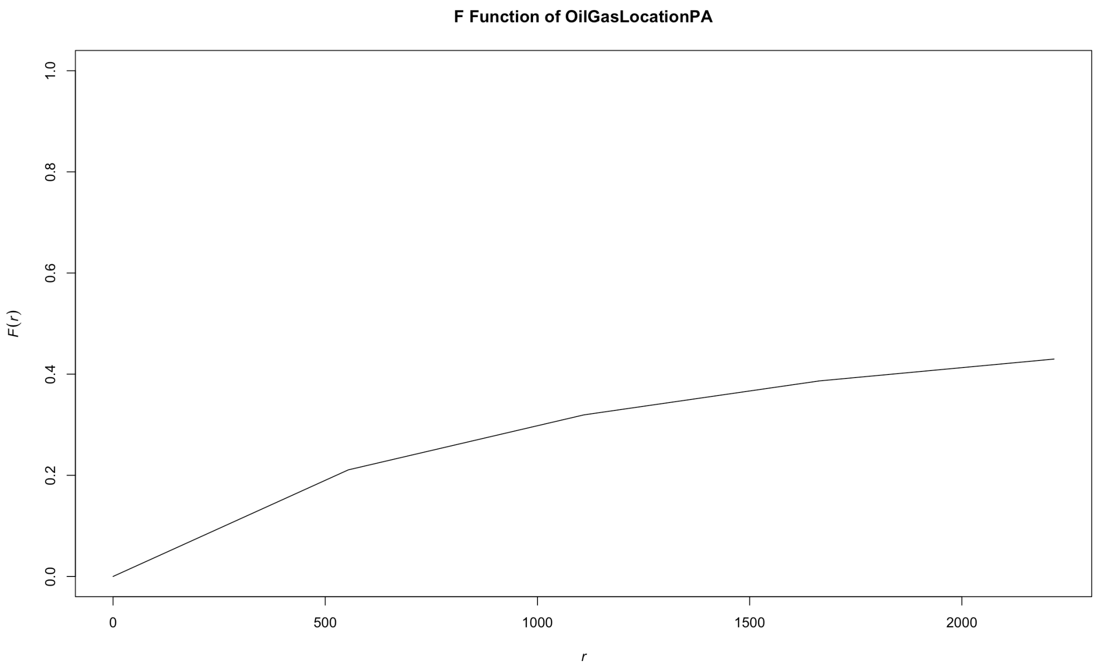
1. Autocorrelation

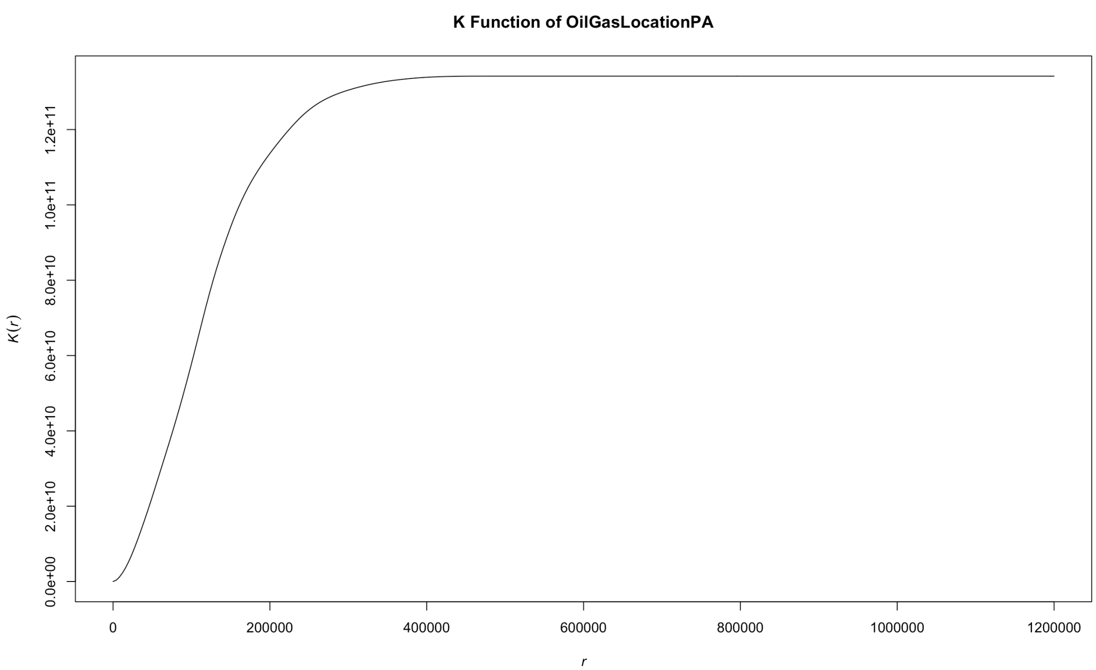
The Moran's I in the attribute Av8top is 0.2265501. Which means, the autocorrelation for attribute Av8top is positive, but not a strong autocorrelation because it's <0.3. More casely speaking, for Av8top, points/areas near each other are similar, but not very strong.

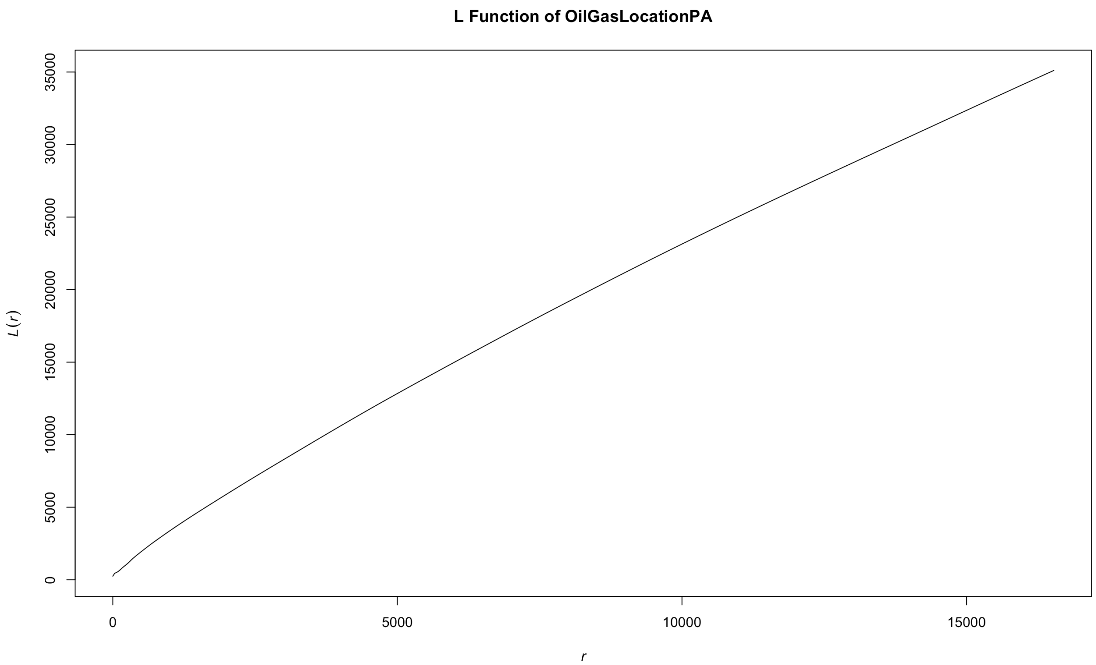
2. Distance-Based Techniques

2.1 G, F, K, L functions plot for OilGasLocationPA



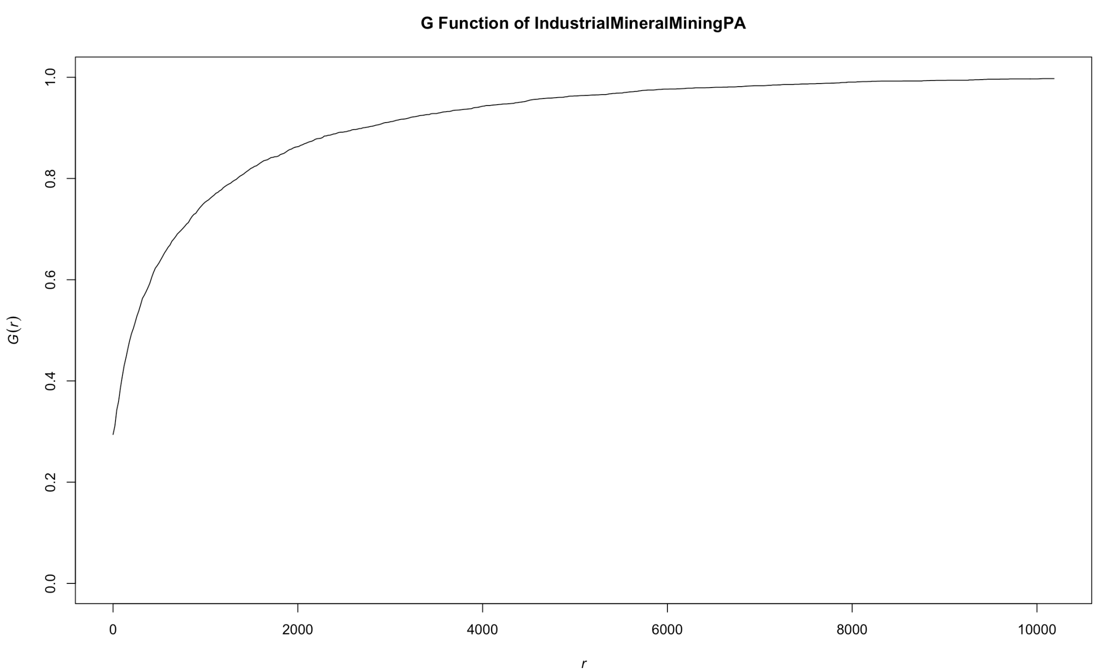

F Function of OilGasLocationPA
F(r)
r
0.0
0.2
0.4
0.6
0.8
1.0
0
500
1000
1500
2000


K Function of OilGasLocationPA
K(r)
r
0.0e+00
2.0e+10
4.0e+10
6.0e+10
8.0e+10
1.0e+11
1.2e+11
0
200000
400000
600000
800000
1000000
1200000

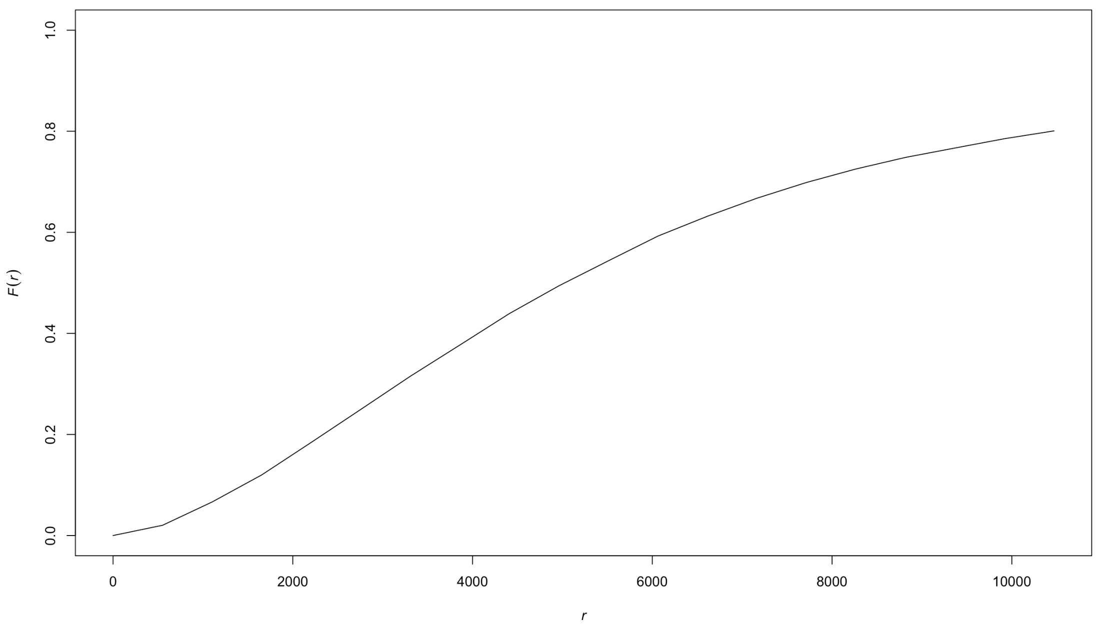


For OilGasLocationPA, G rises sharply at short distances because many event have a very clode nearest neighbor. F rises slowly at first, but more rapidly at longer distances, because a good proportion of the study area is fairly empty. K rises up from 0, through distance 0, and then keep at a level for larger distances, which means clustered. L is above zero, there are more events than expected. So, generally, OilGasLocationPA is clustered.

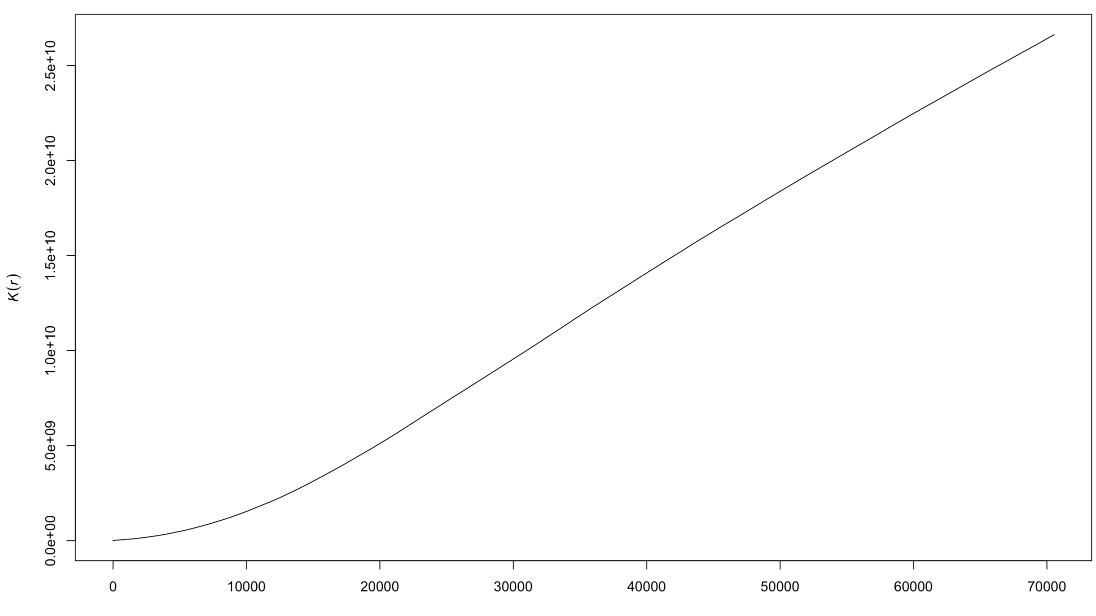
2.2 G, F, K, L functions plot for IndustrialMineralMiningPA

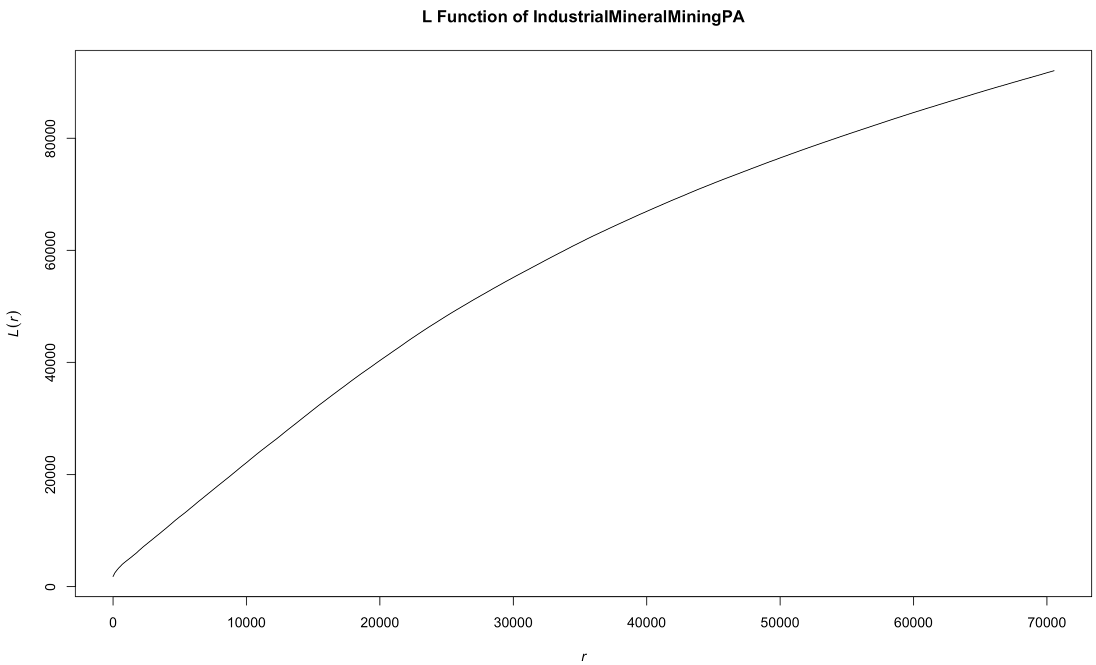


F Function of IndustrialMineralMiningPA



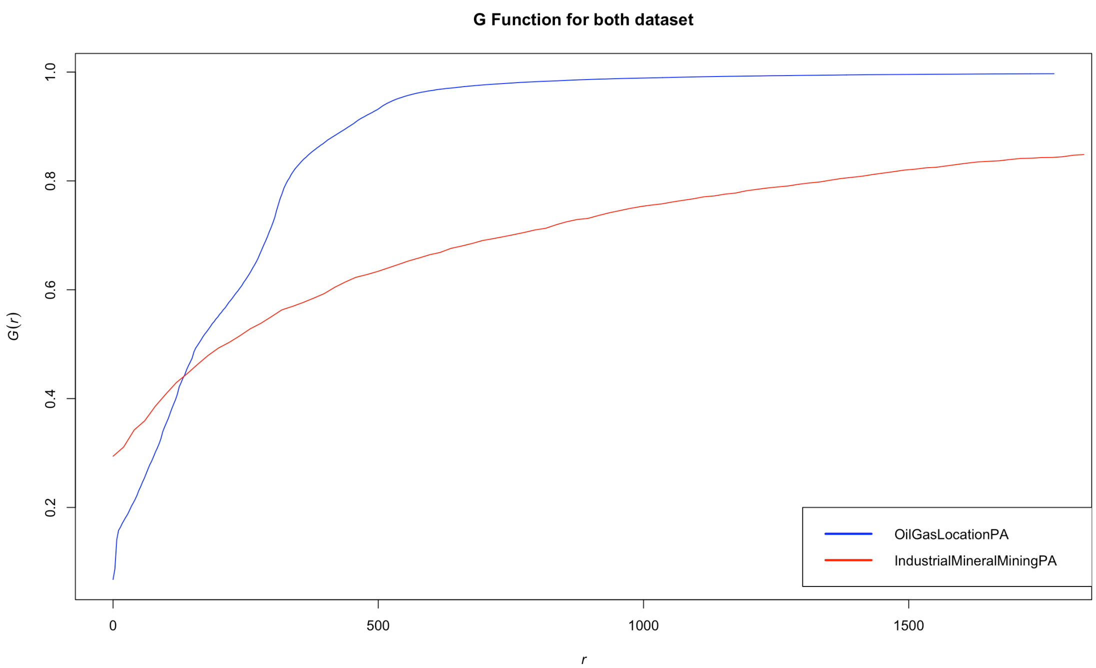
K Function of IndustrialMineralMiningPA





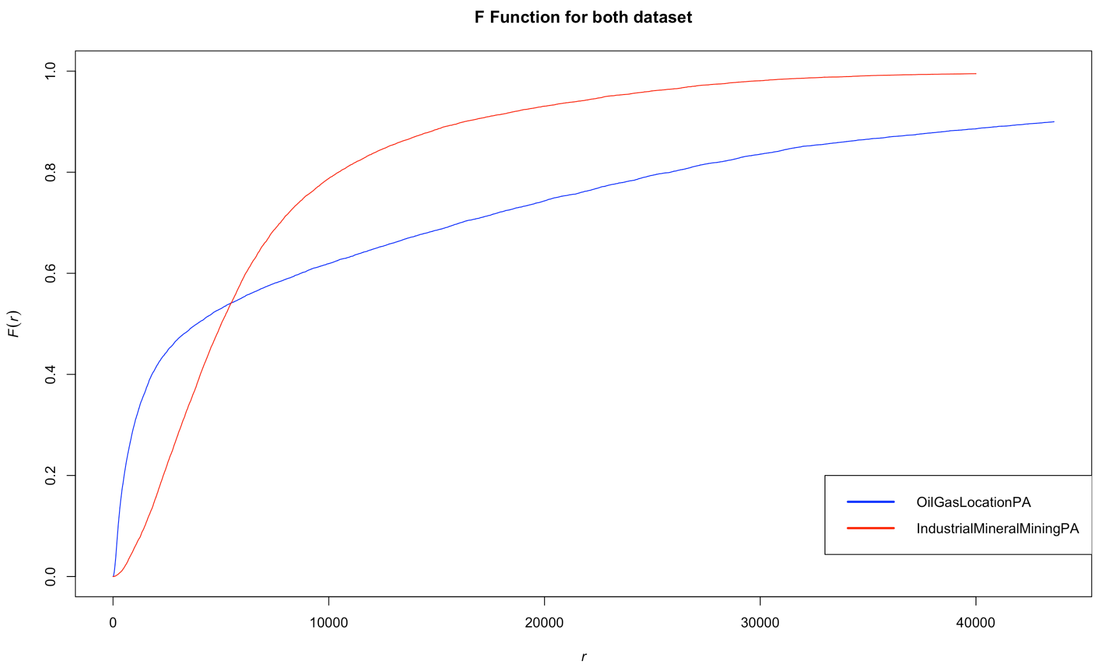
For IndustrialMineralMiningPA, G rises sharply at short distances because many event have a very clode nearest neighbor. F rises slowly at first, but more rapidly at longer distances, because a good proportion of the study area is fairly empty. K rises up from 0, through distance 0. L is above zero, there are more events than expected. So, generally, OilGasLocationPA is clustered.

2.3 Compare G function for both dataset



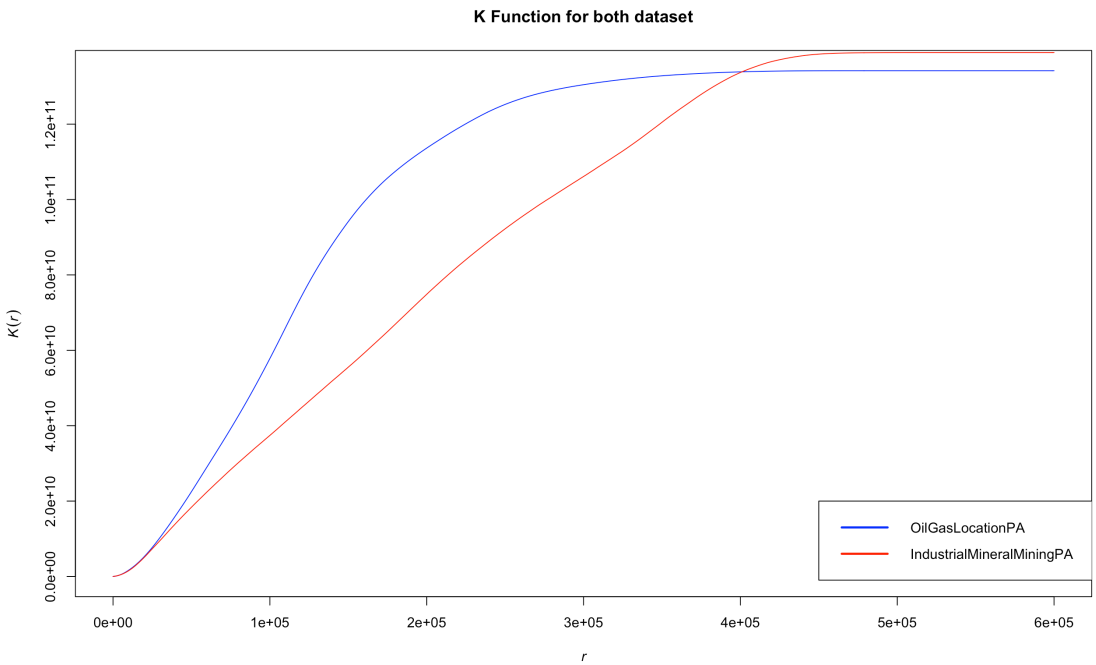
G function for OilGasLocationPA rises more sharply than IndustrialMiningPA. OilGasLocationPA is more clustered than IndustrialMineralMiningPA.

2.4 Compare F function for both dataset



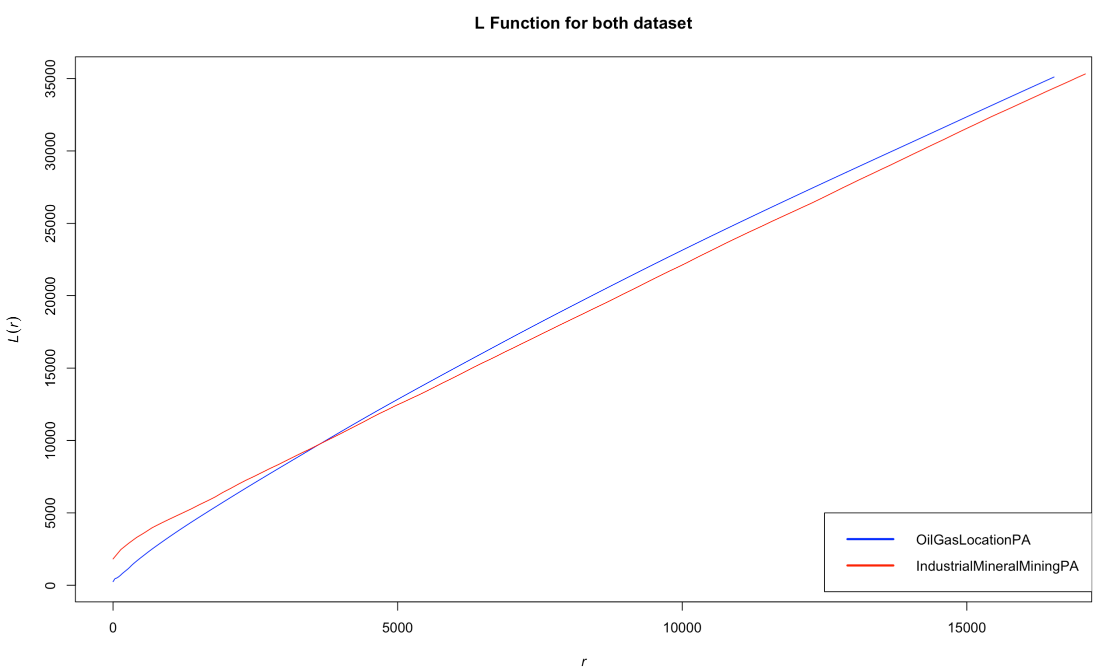
F function for OilGasLocationPA rises quicker than IndustrialMiningPA at short distances, but slower at larger distances. Which means, for OilGasLocationPA, there are more empty area.

2.5 Compare K function for both dataset



At shorter distances, OilGasLocationPA rises quicker, which means there are more points clustered.

2.6 Compare L function for both dataset



For both dataset, L is larger than 0, which means, there are more events than expected. IndustrialMiningPA, at shorter distances (around < 4000), exceed the expectation more than OilGasLocationPA.

3. Code

```
##### 1. Autocorrelation #####
library(ape)  # Analysis of Phylogenetics and Evolution
ozone <- read.csv("ozone.csv", header = TRUE)  # import the ozone.csv file
ozoneLon <- ozone$Lon  # get the longitude of ozone
ozoneLat <- ozone$Lat  # get the latitude of ozone
ozoneDistance <- dist(x = cbind.data.frame(ozoneLon, ozoneLat), method = "euc
lidean")  # calculate the distance using euclidean method
ozoneDistance <- as.matrix(ozoneDistance)  # transform to a matrix
w <- 1/ozoneDistance  # weights w[i,j] = 1/distance[i, j]
diag(w) <- 0  # set the diagnol w[i, i] = 0, instead of Inf
moranI <- Moran.I(x = ozone$Av8top, weight = w, scaled = TRUE)  # calculate M
oran I, scale the result so that it varies between -1 and +1
moranI  # the observed, computed Moran's I is 0.2265501

##### 2. Distance-Based Techniques #####
library(rgdal)  # Bindings for the geospatial data abstraction library.
library(spatstat)  # Spatial Point Pattern Analysis, Model-Fitting, Simulatio
n, Tests
library(maptools)  # Tools for Reading and Handling Spatial Objects

##### 2.1 G, F, K, L functions plot for OilGasLocationPA #####
OilGasLocationPA <- readOGR(dsn = "/Users/qijiawen/Desktop/2017 Spring/Spatia
l Data Analytics/Project 3", layer = "OilGasLocationPA")  # read OilGasLocati
onPA shapefile
OilGasLocationPA.spatialPoints <- as(OilGasLocationPA, "SpatialPoints")  # ch
ange spatial points data frame to Spatial Points
OilGasLocationPA.ppp <- as(OilGasLocationPA.spatialPoints, "ppp")  # change s
patial points to spatial point pattern class ppp

GFunction <- Gest(OilGasLocationPA.ppp, correction = "none")  # G function
FFunction <- Fest(OilGasLocationPA.ppp, correction = "none")  # F function
dk = seq(from = 0, to = 6e+05 * 2, by = 2000 * 2)
KFunction <- Kest(OilGasLocationPA.ppp, correction = "none", r = dk)  # K fun
ction
LFunction <- Lest(OilGasLocationPA.ppp, correction = "none")  # L function
plot(GFunction[, -2], main = "G Function of OilGasLocationPA")
plot(FFunction[, -2], main = "F Function of OilGasLocationPA", ylim = c(0, 1)
)
plot(KFunction[, -2], main = "K Function of OilGasLocationPA")
plot(LFunction[, -2], main = "L Function of OilGasLocationPA")


##### 2.2 G, F, K, L functions plot for IndustrialMineralMiningPA #####
IndustrialMineralMiningPA <- readOGR(dsn = "/Users/qijiawen/Desktop/2017 Spri
ng/Spatial Data Analytics/Project 3", layer = "IndustrialMineralMiningOperati
ons2014_10")  # read IndustrialMineralMiningPA shapefile
```

```
IndustrialMineralMiningPA.spatialPoints <- as(IndustrialMineralMiningPA, "Spa
tialPoints")  # change spatial points data frame to spatial points
IndustrialMineralMiningPA.ppp <- as(IndustrialMineralMiningPA.spatialPoints,
"ppp")  # change spatial points to spatial point pattern class ppp

GFunction_IMMP <- Gest(IndustrialMineralMiningPA.ppp, correction = "none")  #
G function
FFunction_IMMP <- Fest(IndustrialMineralMiningPA.ppp, correction = "none")  #
F function
KFunction_IMMP <- Kest(IndustrialMineralMiningPA.ppp, correction = "none")  #
K function
LFunction_IMMP <- Lest(IndustrialMineralMiningPA.ppp, correction = "none")  #
L function
plot(GFunction_IMMP[, -2], main = "G Function of IndustrialMineralMiningPA",
ylim = c(0, 1))
plot(FFunction_IMMP[, -2], main = "F Function of IndustrialMineralMiningPA",
ylim = c(0, 1))
plot(KFunction_IMMP[, -2], main = "K Function of IndustrialMineralMiningPA")
plot(LFunction_IMMP[, -2], main = "L Function of IndustrialMineralMiningPA")


##### 2.3 Compare G function for both dataset #####
plot(GFunction[, -2], main = "G Function for both dataset", col = "blue")
plot(GFunction_IMMP[, -2], col = "red", add = TRUE)
legend(1300, 0.2, c("OilGasLocationPA", "IndustrialMineralMiningPA"), lty = c
(1, 1), lwd = c(2.5, 2.5), col = c("blue", "red"))  # add a legend

##### 2.4 Compare F function for both dataset #####
plot(FFunction[, -2], main = "F Function for both dataset", col = "blue", yli
m = c(0, 1))
d = seq(from = 0, to = 40000, by = 10)
FFunction_IMMP <- Fest(IndustrialMineralMiningPA.ppp, correction = "none", r
= d)
plot(FFunction_IMMP[, -2], col = "red", add = TRUE, ylim = c(0, 1))
legend(33000, 0.2, c("OilGasLocationPA", "IndustrialMineralMiningPA"), lty =
c(1, 1), lwd = c(2.5, 2.5), col = c("blue", "red"))  # add a legend

##### 2.5 Compare K function for both dataset #####
d = seq(from = 0, to = 6e+05, by = 1200)
KOil <- Kest(OilGasLocationPA.ppp, correction = "none", r = d)
KInd <- Kest(IndustrialMineralMiningPA.ppp, correction = "none", r = d)
plot(KOil[, -2], main = "K Function for both dataset", col = "blue")
plot(KInd[, -2], col = "red", add = TRUE)
legend(450000, 2e+10, c("OilGasLocationPA", "IndustrialMineralMiningPA"), lty
= c(1, 1), lwd = c(2.5, 2.5), col = c("blue", "red"))  # add a legend

##### 2.6 Compare L function for both dataset #####
plot(LFunction[, -2], main = "L Function for both dataset", col = "blue")
plot(LFunction_IMMP[, -2], col = "red", add = TRUE)
legend(12500, 5000, c("OilGasLocationPA", "IndustrialMineralMiningPA"), lty =
c(1, 1), lwd = c(2.5, 2.5), col = c("blue", "red"))  # add a legend
```