Group_3_FInal_Project

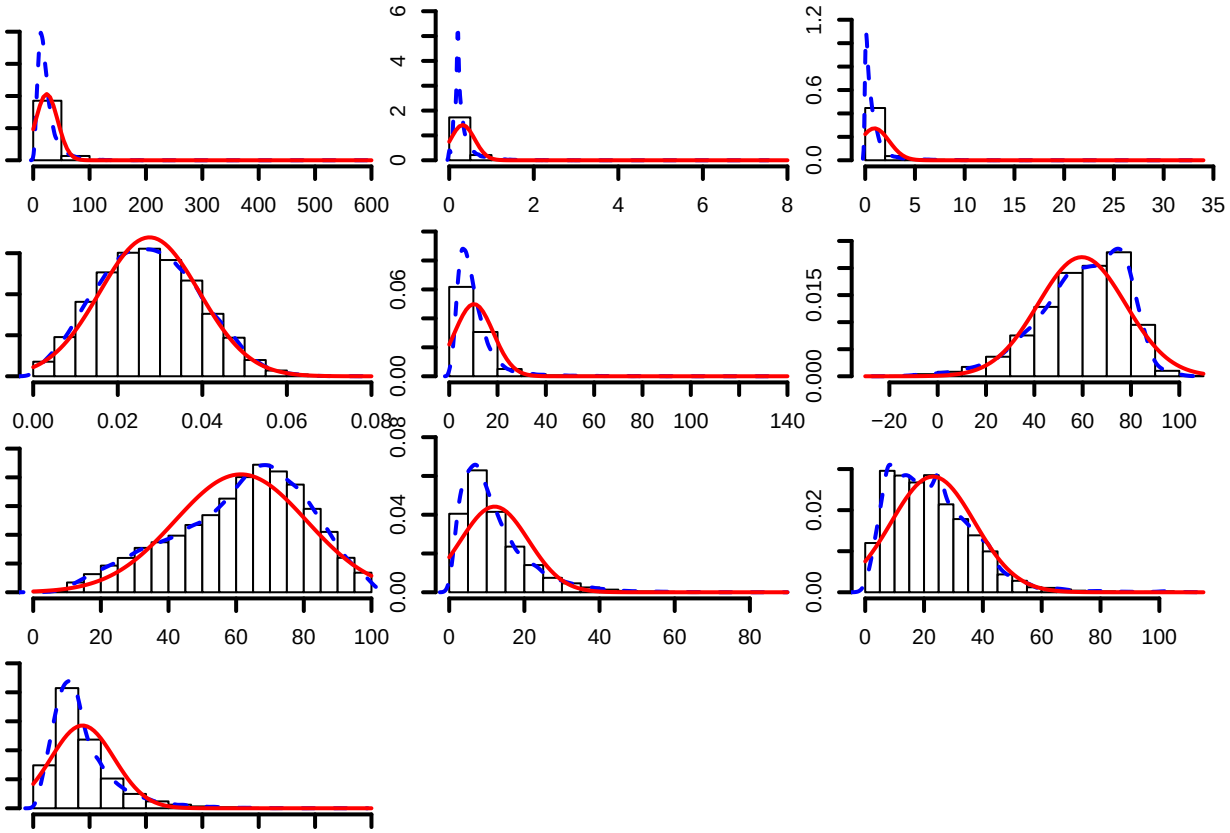
```
# CLear all data
rm(list = ls())
```

```
# Install packages
ipak <- function(pkg) {
  # Install and load multiple R packages
  # Check to see if packages are installed
  # Install them if they are not, then load them
  # Args:
  #  pkg: packaged to be loaded into the R session or installed if not already
  # installed
  # Returns:
  #  Library load messages
  # Check to see if package has been installed
  new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
  # If not installed, then install
  if (length(new.pkg)) {
    install.packages(new.pkg, dependencies = TRUE)
  }
  sapply(pkg, require, character.only = TRUE)
  sapply(pkg, library, character.only = TRUE)
}

#packages to install
pkg <- c("tidyverse", "data.table", "knitr", 'e1071','car','DAAG','corrplot','cluster','NbClust','caret
ipak(pkg)
```

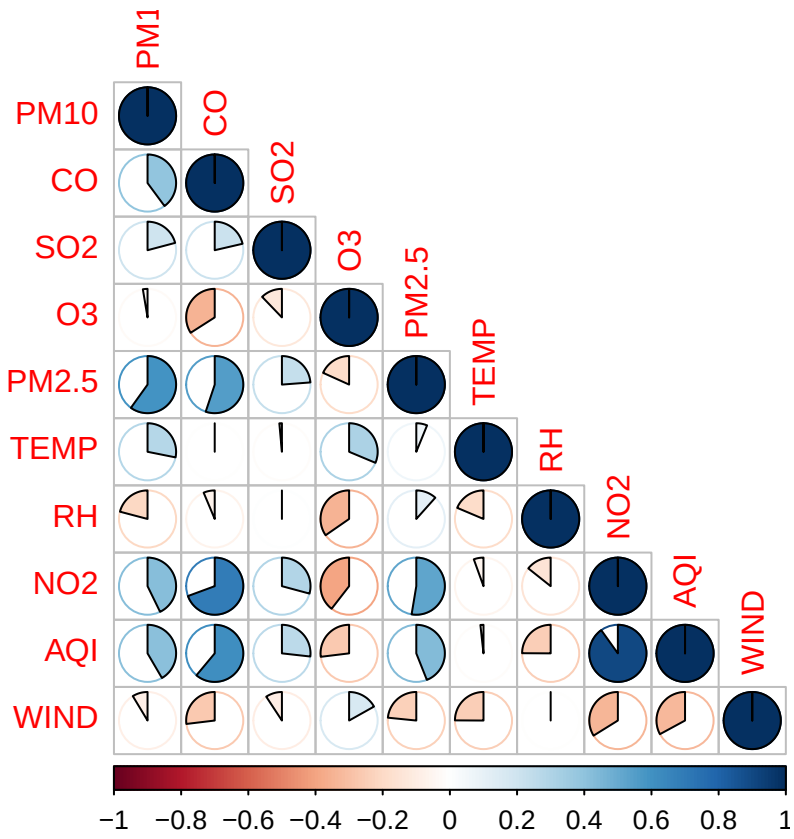
```
library(tidyverse)
library(data.table)
library(e1071)
library(reshape2)
library(plyr)
library(psych)
library(gridExtra)
library(dbscan)
library(fiftystater)
library(factoextra)
library(cluster)
library(cowplot)
library(fpc)
library(mclust)
library(corrplot)
library(NbClust)
library(caret)
library(MASS)
library(car)
library(DAAG)
```

```
# Make sure file is in same directory otherwise edit this section if it's not
prj_data <- read.csv("/Users/satishreddychirra/Document/aqi_new_nd.csv", header = T)
```

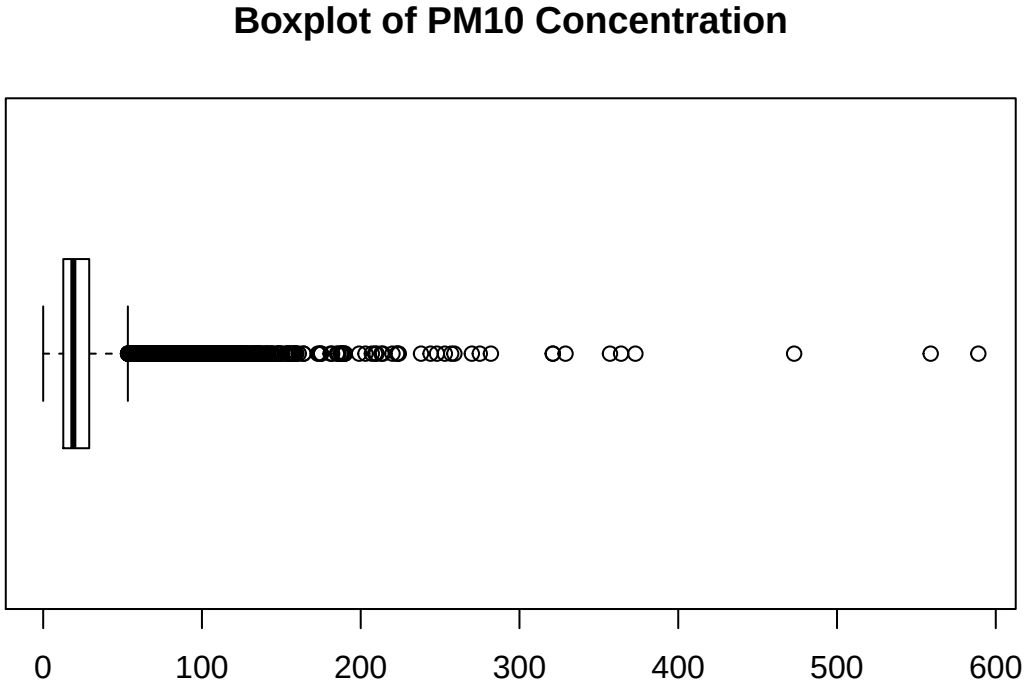
```r
# Only reading the columns used for histograms
histdata <- prj_data
histdata <- subset(histdata, select = pm10_arithmetic_mean_n:wind_arithmetic_mean_n)
names(histdata)[1]<-"PM10"
names(histdata)[2]<-"CO"
names(histdata)[3]<-"SO2"
names(histdata)[4]<-"O3"
names(histdata)[5]<-"PM2.5"
names(histdata)[6]<-"TEMP"
names(histdata)[7]<-"RH"
names(histdata)[8]<-"NO2"
names(histdata)[9]<-"AQI"
names(histdata)[10]<-"WIND"

par(mar=c(1,1,1,1))
multi.hist(histdata, ylab=" ",density=TRUE, cex.lab=1.7, bcol="white",
       dcol=c("blue","red"),dlty=c("dashed","solid"), lwd=2 ,main= " ",freq=FALSE)
```



```r
# Plotting the intecorrelations
M <- cor(histdata, method="pearson")
corrplot(M,"pie", "lower")
```

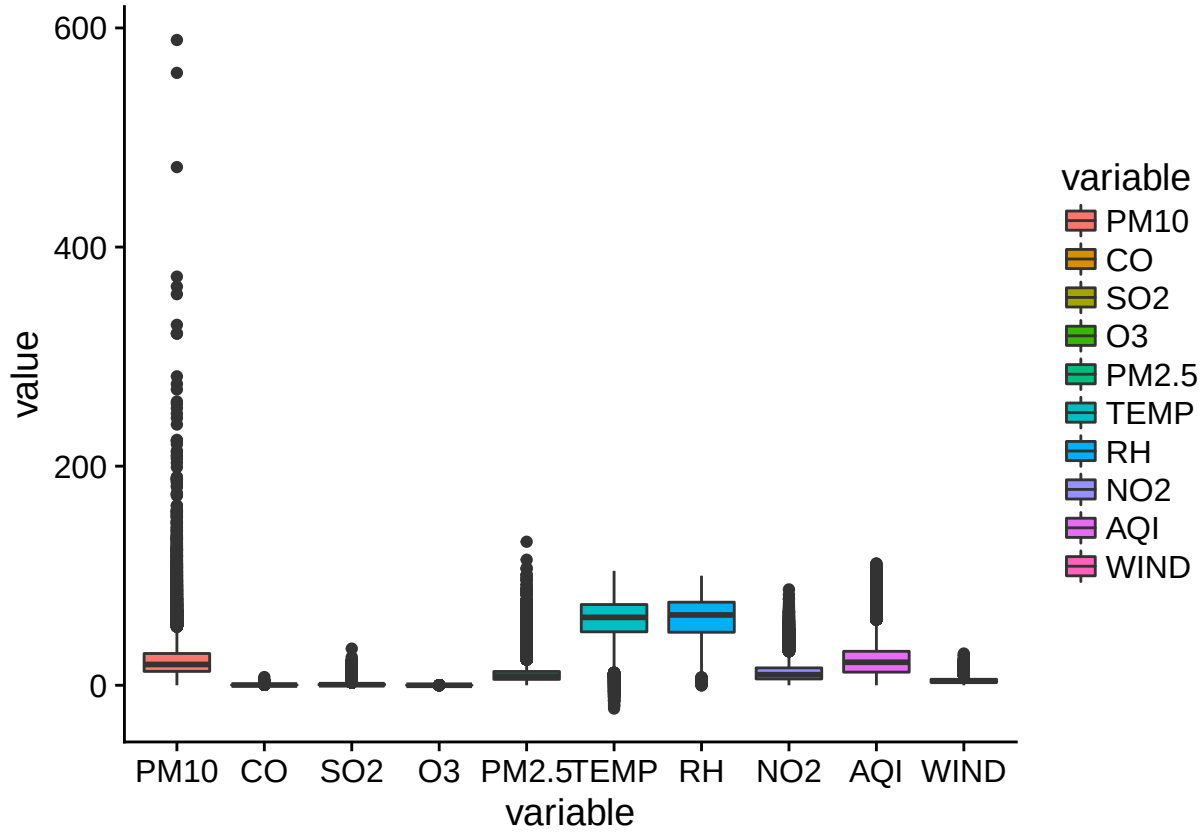



```
# We then use boxplots to further identify the outliers
boxplot(histdata$PM10,horizontal=TRUE,main="Boxplot of PM10 Concentration")
```




```
boxplo1 <- ggplot(data = melt(histdata), aes(x=variable, y=value)) + geom_boxplot(aes(fill=variable))

## No id variables; using all as measure variables
```

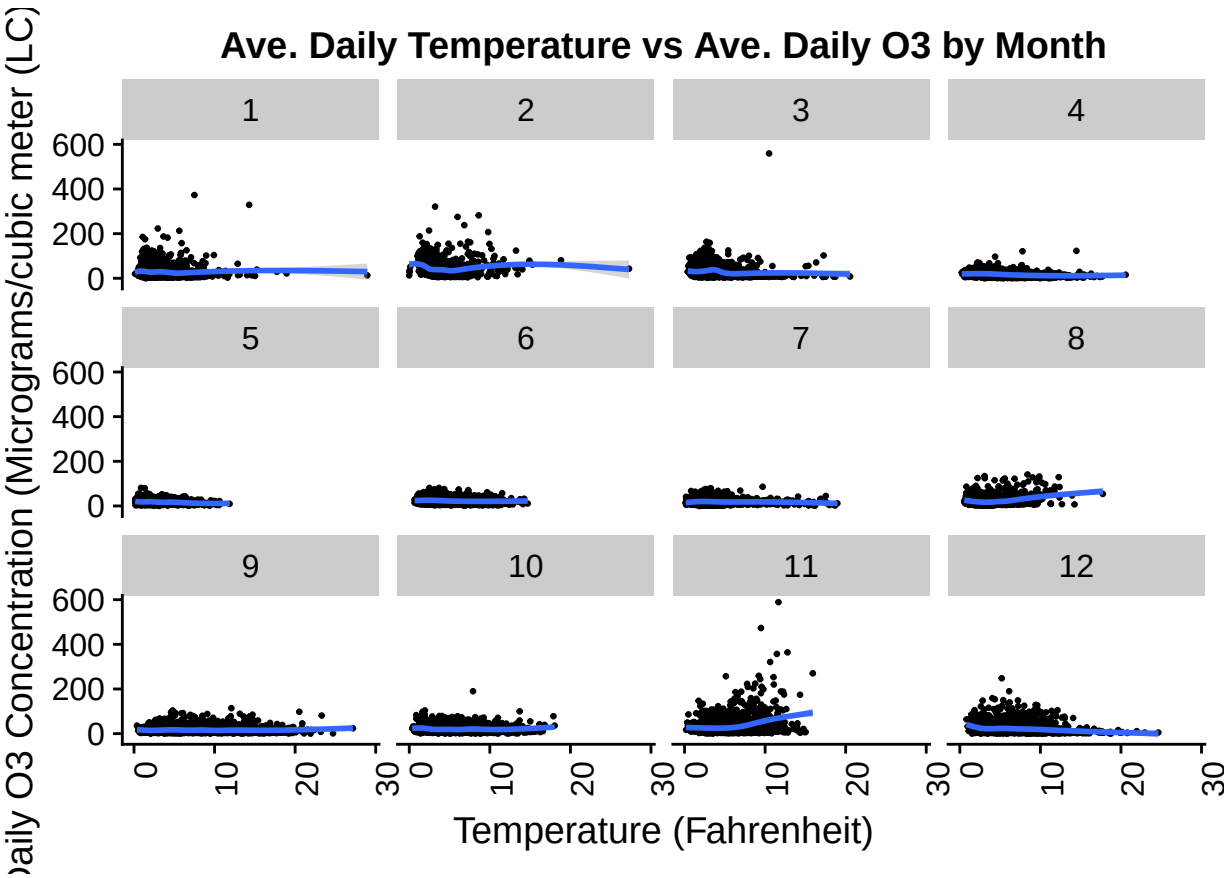
```
boxplo1
```




```
# Creating categorical variables for the seasons
prj_data$spring <- ifelse(prj_data$month == "3" ,1, ifelse(prj_data$month == "4", 1, ifelse(prj_data$mon

prj_data$summer <- ifelse(prj_data$month == "6" ,1, ifelse(prj_data$month == "7", 1, ifelse(prj_data$mon

prj_data$fall <- ifelse(prj_data$month == "9" ,1, ifelse(prj_data$month == "10", 1, ifelse(prj_data$mont

prj_data$winter <- ifelse(prj_data$month == "12" ,1, ifelse(prj_data$month == "1", 1, ifelse(prj_data$m
```

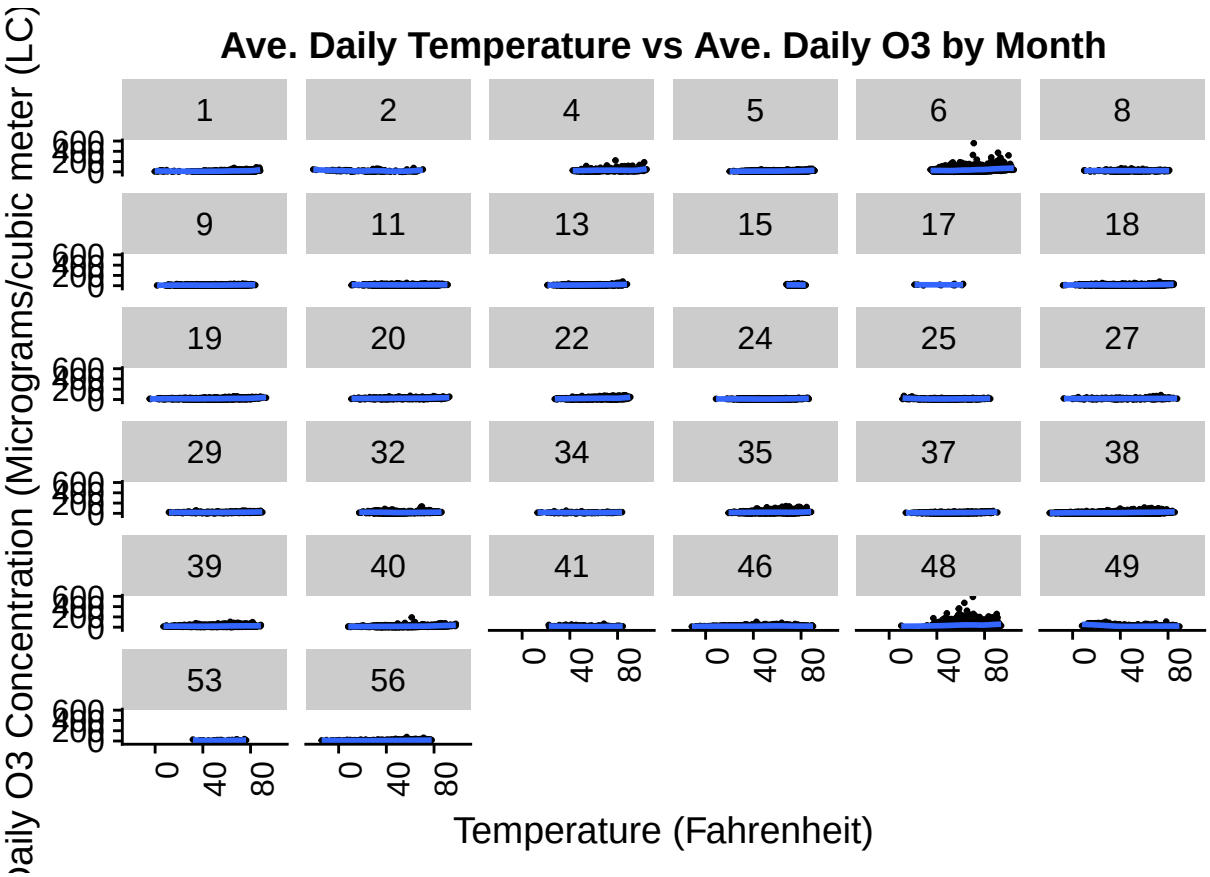
```
# Comparing different months
p1 <- ggplot(prj_data, aes(x = prj_data$wind_arithmetic_mean_n, y = prj_data$pm10_arithmetic_mean_n)) +
print(p1)
```

```
## `geom_smooth()` using method = 'gam'
```




```
# Comparing different States
p2 <- ggplot(prj_data, aes(x = prj_data$temp_arithmetic_mean_n, y = prj_data$pm10_arithmetic_mean_n)) +
print(p2)
```

```
## `geom_smooth()` using method = 'gam'
```




```
# Obtaining the final set before the clustering
total_data <- subset(prj_data, select = pm10_arithmetic_mean_n:wind_arithmetic_mean_n)
total_data <- cbind(total_data,factor(prj_data$spring),factor(prj_data$summer),factor(prj_data$fall),fa
total_data$no2_aqi_n <- NULL
```

```
# Scale the data
fn <- function(x) x * 1/max(x, na.rm = TRUE)
ind <- sapply(total_data, is.numeric)
total_data[ind] <- lapply(total_data[ind], fn)
```

```
glimpse(total_data)
```

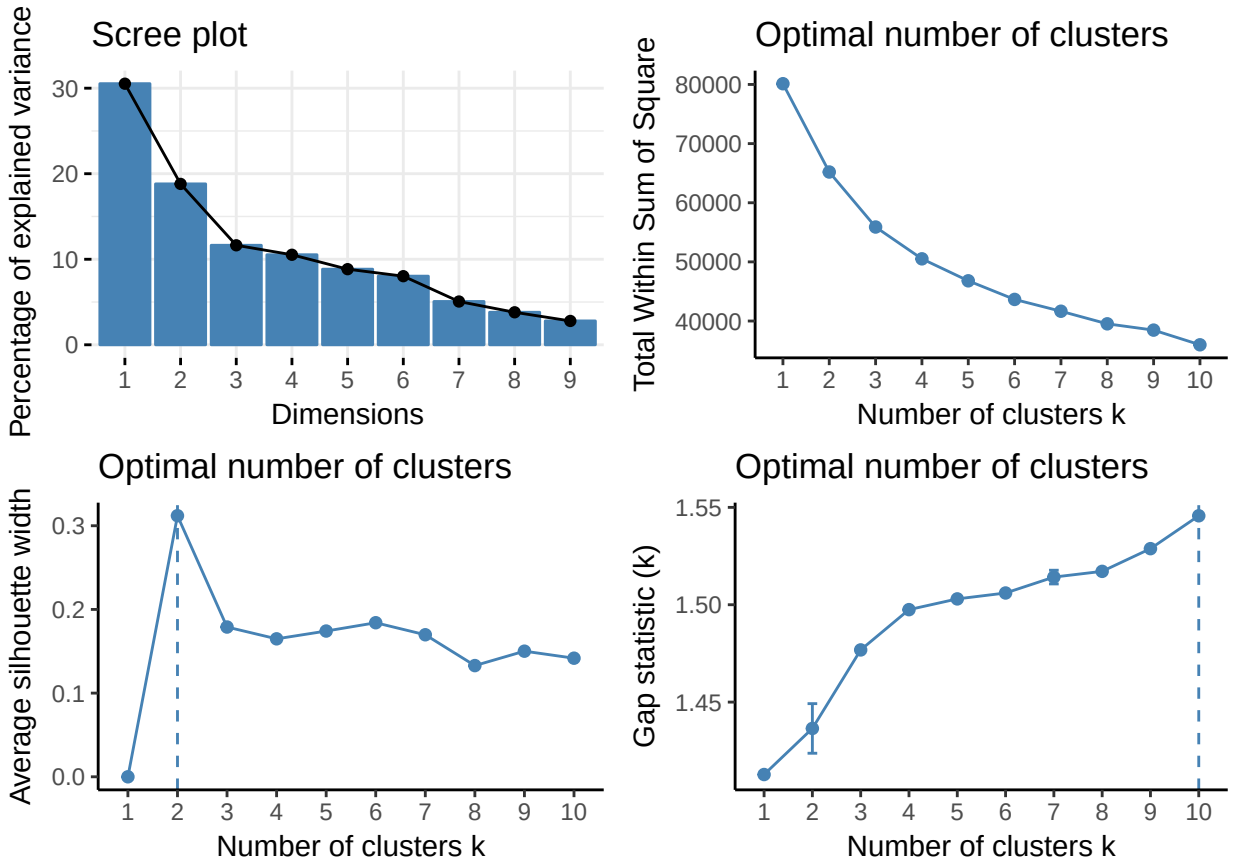
```
## Observations: 36,712
## Variables: 13
## $ pm10_arithmetic_mean_n    <dbl> 0.023769100, 0.016977929, 0.02037351...
## $ co_arithmetic_mean_n      <dbl> 0.021087637, 0.021642620, 0.01331853...
## $ so2_arithmetic_mean_n     <dbl> 0.027035961, 0.015478219, 0.00708407...
## $ o3_arithmetic_mean_n      <dbl> 0.1608620, 0.2467164, 0.2211967, 0.1...
## $ pm25f_arithmetic_mean_n   <dbl> 0.07633588, 0.06030534, 0.08676845, ...
## $ temp_arithmetic_mean_n    <dbl> 0.017550854, 0.099920224, 0.02197845...
## $ rhdp_arithmetic_mean_n    <dbl> 0.5325000, 0.8362500, 0.7720833, 0.6...
## $ no2_arithmetic_mean_n     <dbl> 0.13834658, 0.10449784, 0.04308453, ...
## $ wind_arithmetic_mean_n    <dbl> 0.07826213, 0.08487986, 0.06991800, ...
## $ `factor(prj_data$spring)` <fctr> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ `factor(prj_data$summer)` <fctr> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ `factor(prj_data$fall)`   <fctr> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ `factor(prj_data$winter)` <fctr> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
```

```r
# Subsetting the data based on seasons
data_summer <- prj_data[ which(prj_data$summer=='1'), ]
data_spring <- prj_data[ which(prj_data$spring=='1'), ]
data_fall <- prj_data[ which(prj_data$fall=='1'), ]
data_winter <- prj_data[ which(prj_data$winter=='1'), ]
```

Summer Data

```r
# Extracting the relevant variables and then scale them to have a mean of 0 and std with 1
data_sum_n <- subset(data_summer, select = pm10_arithmetic_mean_n:wind_arithmetic_mean_n)
data_sum_n <- subset(data_sum_n, select = -no2_aqi_n)

names(data_sum_n)[1]<-"PM10"
names(data_sum_n)[2]<-"CO"
names(data_sum_n)[3]<-"SO2"
names(data_sum_n)[4]<-"O3"
names(data_sum_n)[5]<-"PM2.5"
names(data_sum_n)[6]<-"TEMP"
names(data_sum_n)[7]<-"RH"
names(data_sum_n)[8]<-"NO2"
names(data_sum_n)[9]<-"WIND"
scaled_summ_data <- scale(data_sum_n)
```

```r
# Using Kmeans finding K
theme_set(theme_cowplot(font_size=10)) # reduce default font size
summer_data.pca <- prcomp(scaled_summ_data,center = FALSE, scale = FALSE)
w1<- fviz_eig(summer_data.pca)
w2 <- fviz_nbclust(scaled_summ_data, kmeans, method = "wss") +
  theme_classic()
w3 <- fviz_nbclust(scaled_summ_data, kmeans, method = "silhouette") +
  theme_classic()
gap_statsu <- clusGap(scaled_summ_data, FUN = kmeans, K.max = 10, B = 2)


# Plot gap statistic
w4 <- fviz_gap_stat(gap_statsu) +
  theme_classic()

plot_grid(w1,w2,w3,w4)
```



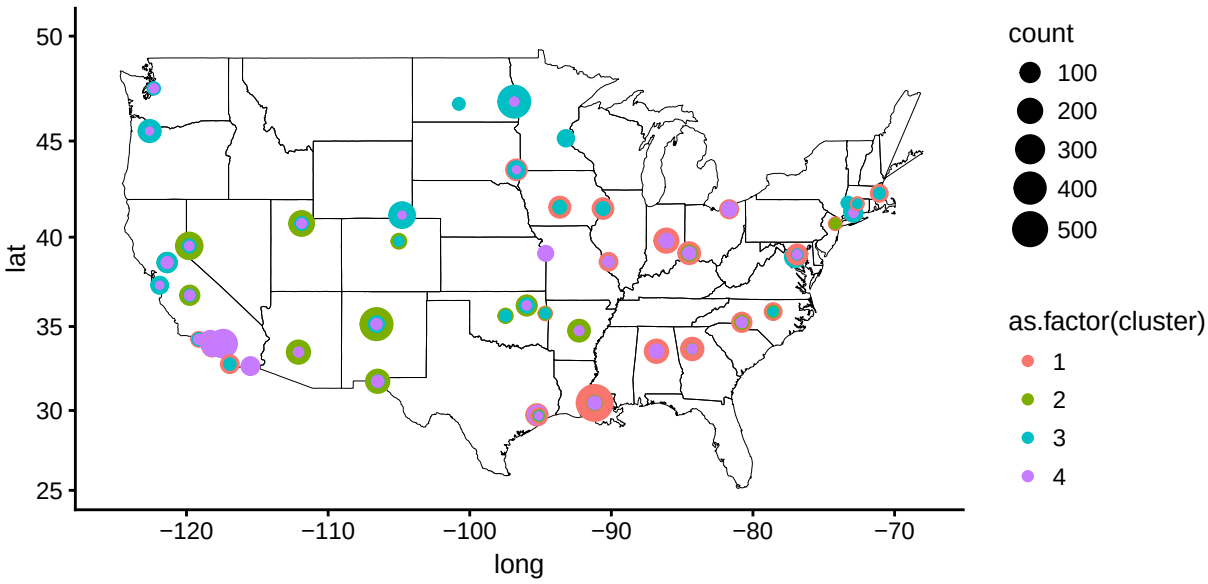
K-Means on summer data

```
set.seed(123)
# Taking k as 4
fit_summ <- kmeans(scaled_summ_data, 4)

# Assigning cluster to the dataset
data_sum_n$cluster <- fit_summ$cluster
data_summer$cluster <- fit_summ$cluster

# Plotting on the geolocations
summ_plot <-data_summer %>%
  dplyr::select(pm10_latitude_n,pm10_longitude_n,cluster)

summ_plot_n <- summ_plot %>%
   dplyr::count(pm10_longitude_n, pm10_latitude_n, cluster)

colnames(summ_plot_n) <- c("pm10_longitude_n","pm10_latitude_n","cluster","count")

statesUSA <- map_data("state")
```

```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:DAAG':
##
##     ozone

## The following object is masked from 'package:mclust':
##
##     map

## The following object is masked from 'package:cluster':
##
##     votes.repub

## The following object is masked from 'package:plyr':
##
##     ozone

## The following object is masked from 'package:purrr':
##
##     map
```

```
ggplot(data = statesUSA) +
  geom_polygon(aes(x = long, y = lat, group = group), fill = NA, color = "black",size = 0.15) +
  coord_map() +
  geom_point(data = summ_plot_n, aes(x = pm10_longitude_n, y = pm10_latitude_n, color = as.factor(cluste
  scale_x_continuous(limits = c(-125,-68)) +
  scale_y_continuous(limits = c(25,50))
```

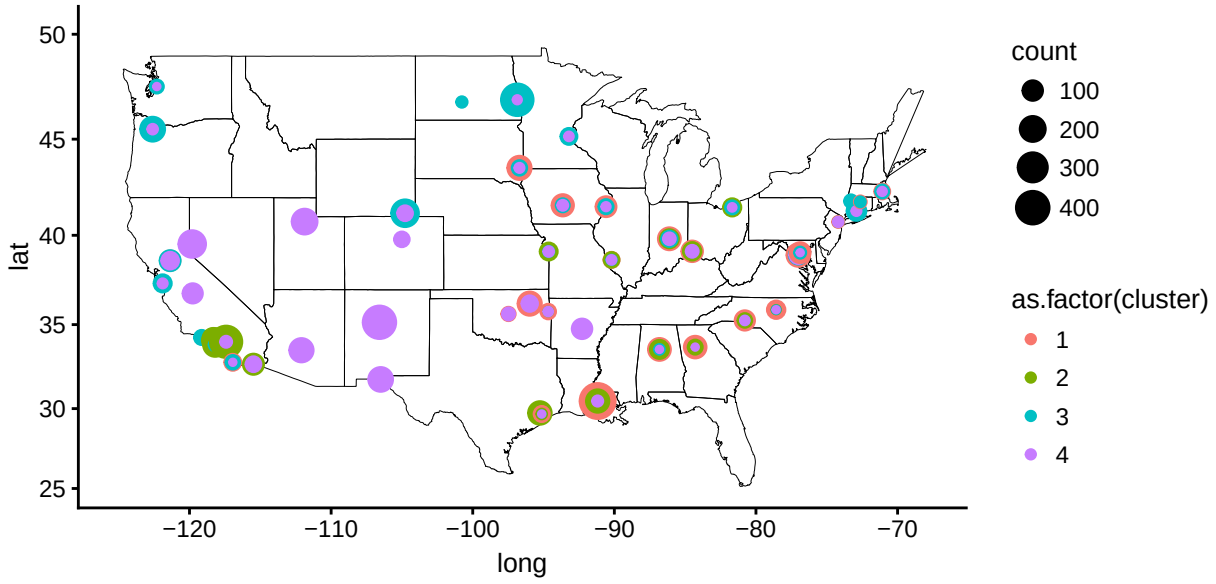



```
cesu <- aggregate(data_sum_n, by=list(cluster=fit_summ$cluster), mean)
is.num <- sapply(cesu, is.numeric)
cesu[is.num] <- lapply(cesu[is.num], round, 2)
cesu[-11]
```

```
##   cluster  PM10   CO  SO2   O3 PM2.5  TEMP    RH   NO2 WIND
## 1       1 22.82 0.22 0.70 0.03 10.00 77.20 71.57  8.10 2.84
## 2       2 26.20 0.21 0.65 0.04  9.00 81.36 37.32  8.69 3.92
## 3       3 14.59 0.15 0.41 0.03  5.89 68.57 66.11  4.71 6.31
## 4       4 55.52 0.47 2.56 0.03 20.81 78.46 61.91 19.91 3.75
```

Partition Around Medoids on Summer Data

```
set.seed(123)
# Taking k as 4
fit_summ <- clara(scaled_summ_data, 4, medoids.x = TRUE)

# Assigning cluster to the dataset
data_sum_n$cluster <- fit_summ$cluster
data_summer$cluster <- fit_summ$cluster

# Plotting on the geolocations
summ_plot <-data_summer %>%
  dplyr::select(pm10_latitude_n,pm10_longitude_n,cluster)

summ_plot_n <- summ_plot %>%
   dplyr::count(pm10_longitude_n, pm10_latitude_n, cluster)

colnames(summ_plot_n) <- c("pm10_longitude_n","pm10_latitude_n","cluster","count")

ggplot(data = statesUSA) +
  geom_polygon(aes(x = long, y = lat, group = group), fill = NA, color = "black",size = 0.15) +
  coord_map() +
  geom_point(data = summ_plot_n, aes(x = pm10_longitude_n, y = pm10_latitude_n, color = as.factor(cluste
  scale_x_continuous(limits = c(-125,-68)) +
  scale_y_continuous(limits = c(25,50))
```



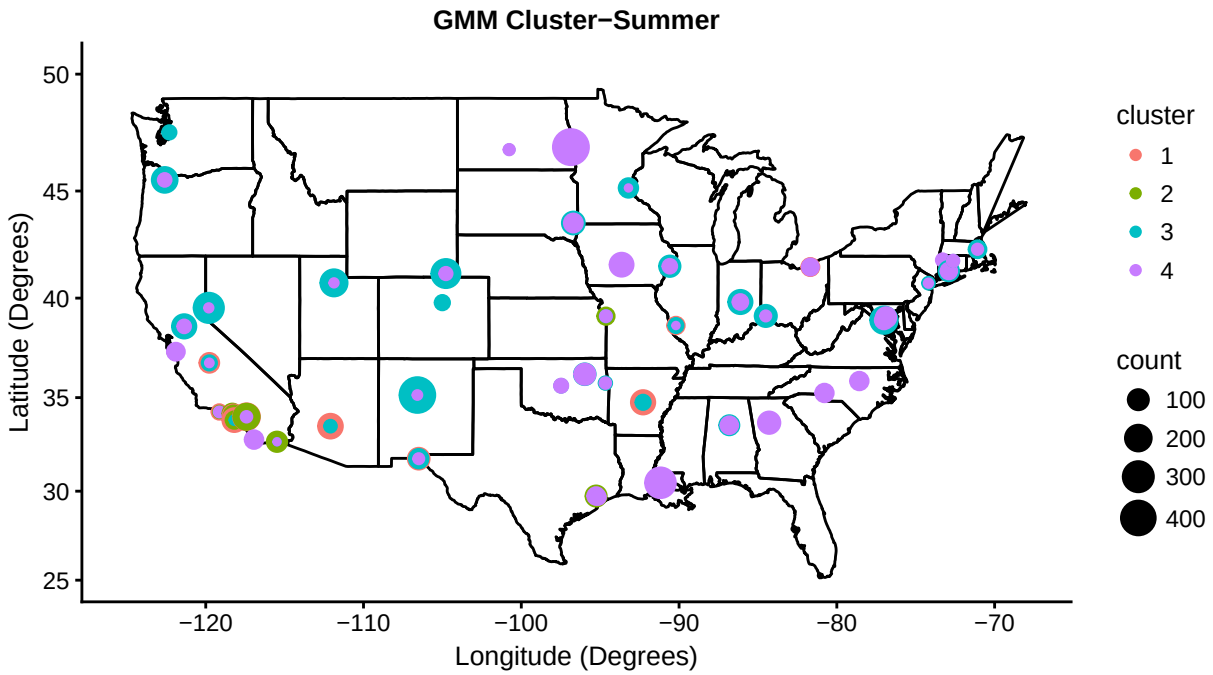

```
t = data.frame(round(fit_summ$medoids, 4))

t$PM10_n <- c((t$PM10*sd(data_summer$pm10_arithmetic_mean_n))+ mean(data_summer$pm10_arithmetic_mean_n)
t$CO_n <- c((t$CO*sd(data_summer$co_arithmetic_mean_n))+ mean(data_summer$co_arithmetic_mean_n))
t$SO2_n <- c((t$SO2*sd(data_summer$so2_arithmetic_mean_n))+ mean(data_summer$so2_arithmetic_mean_n))
t$O3_n <- c((t$O3*sd(data_summer$o3_arithmetic_mean_n))+ mean(data_summer$o3_arithmetic_mean_n))
t$PM2.5_n <- c((t$PM2.5*sd(data_summer$pm25f_arithmetic_mean_n))+ mean(data_summer$pm25f_arithmetic_mean
t$TEMP_n <- c((t$TEMP*sd(data_summer$temp_arithmetic_mean_n))+ mean(data_summer$temp_arithmetic_mean_n)
t$RH_n <- c((t$RH*sd(data_summer$rhdp_arithmetic_mean_n))+ mean(data_summer$rhdp_arithmetic_mean_n))
```

```r
t$NO2_n <- c((t$NO2*sd(data_summer$no2_arithmetic_mean_n))+ mean(data_summer$no2_arithmetic_mean_n))
t$WIND_n <- c((t$WIND*sd(data_summer$wind_arithmetic_mean_n))+ mean(data_summer$wind_arithmetic_mean_n))

med_new <- t %>% dplyr::select(PM10_n:WIND_n)

med_new
```

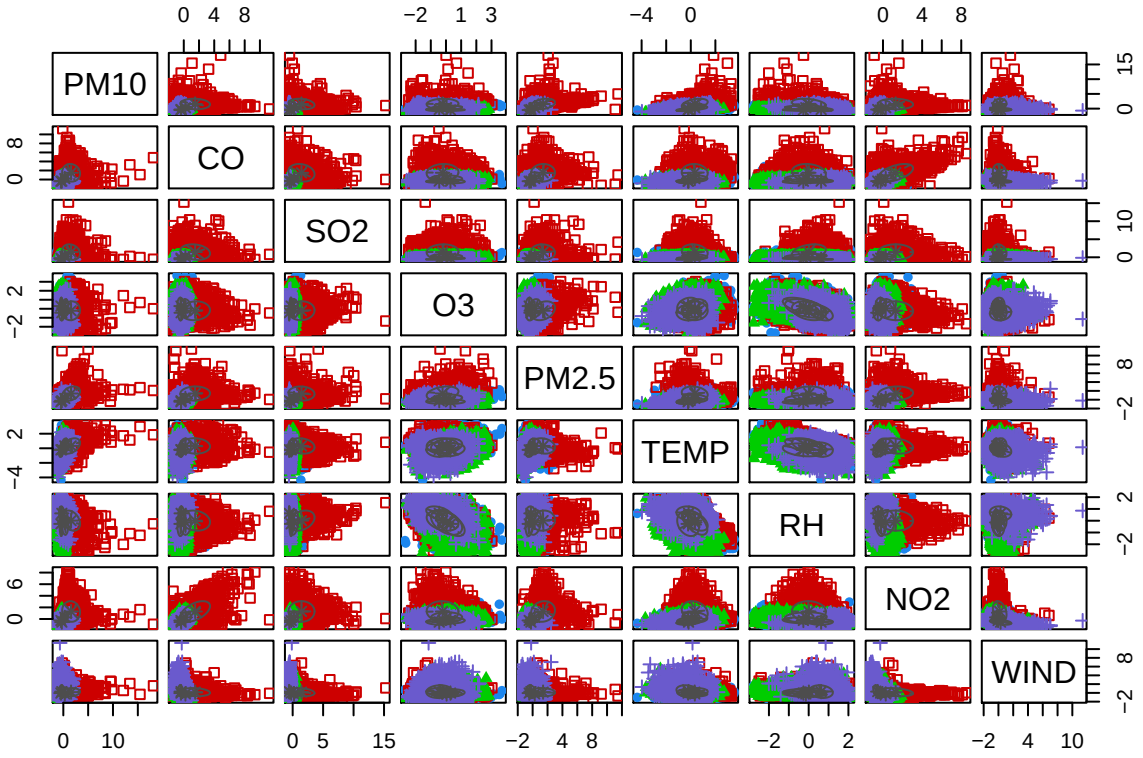
```
##         PM10_n      CO_n     SO2_n       O3_n   PM2.5_n   TEMP_n     RH_n
## 31765 25.00066 0.1374957 0.3999672 0.03394138  7.250198 78.79162 65.75047
## 20854 43.00063 0.3333254 1.9833407 0.03988241 16.050313 82.49964 70.00046
## 31240 11.50020 0.1416712 0.1020686 0.03482349  3.299948 63.12494 61.54243
## 23235 30.00076 0.1958399 0.9895327 0.04941178  9.300236 78.83320 30.58308
##          NO2_n   WIND_n
## 31765  2.837264 2.566663
## 20854 13.074791 2.383368
## 31240  5.804454 4.491590
## 23235 11.241458 3.349892
```

GMM on Summer Data

```r
set.seed(123)
# Fit GMM - G = 4 (4 clusters forced to compare to K-means)
fit_summ_d <- Mclust(scaled_summ_data, G = 4)

# Assigning cluster to the dataset
data_summer$cluster <- fit_summ_d$classification
summ_plot <- data.frame(
  data_summer$pm10_longitude_n,
  data_summer$pm10_latitude_n,
  data_summer$cluster
)
colnames(summ_plot) <- c("pm10_longitude_n","pm10_latitude_n","cluster")

summ_plot_n_dens <- summ_plot %>%
  dplyr::count(pm10_longitude_n, pm10_latitude_n, cluster)

colnames(summ_plot_n_dens) <- c("pm10_longitude_n","pm10_latitude_n","cluster","count")

summ_plot_n_dens$cluster <- as.factor(summ_plot_n_dens$cluster)

ggplot(data = statesUSA) +
  geom_polygon(aes(x = long, y = lat, group = group), fill = NA, color = "black") +
  coord_map() +
  geom_point(data = summ_plot_n_dens, aes(x = pm10_longitude_n, y = pm10_latitude_n, color = cluster, si
  scale_x_continuous(limits = c(-125,-68)) +
  scale_y_continuous(limits = c(25,50)) +
  labs(title = "GMM Cluster-Summer", x = 'Longitude (Degrees)', y = 'Latitude (Degrees)')
```



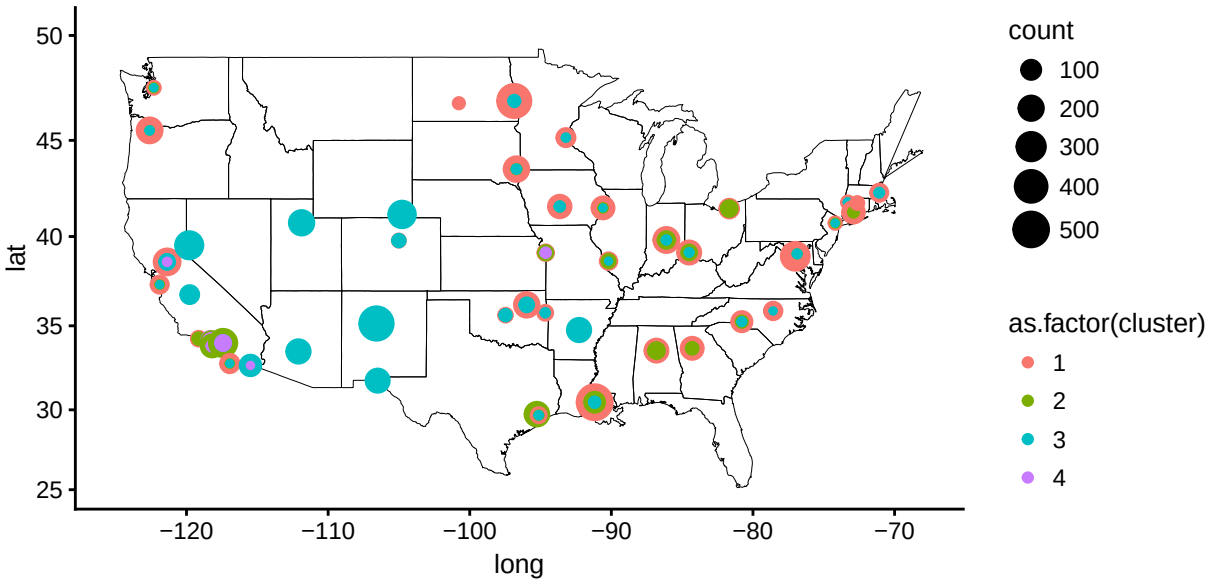

```
plot(fit_summ_d, what = "classification")
```



Hierarchical Clustering on Summer Data

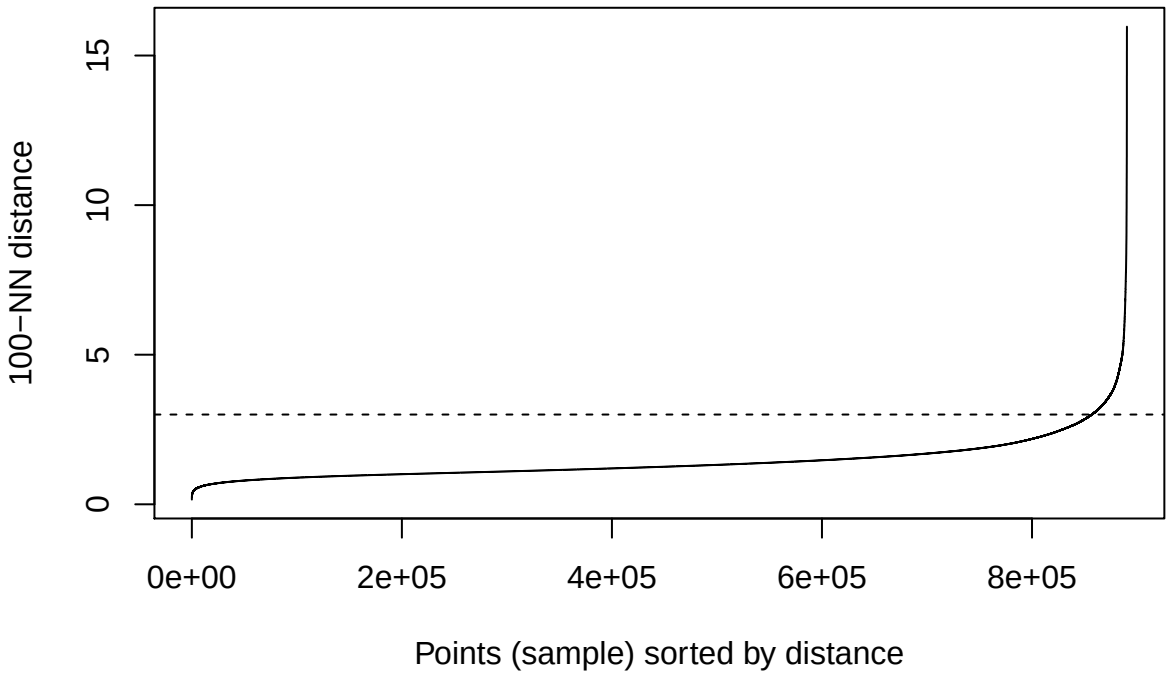
```
set.seed(123)
```

```
# Taking k as 4
fit_summ <- hcut(scaled_summ_data, 4)

# Assigning cluster to the dataset
data_sum_n$cluster <- fit_summ$cluster
data_summer$cluster <- fit_summ$cluster

# Plotting on the geolocations
summ_plot <-data_summer %>%
  dplyr::select(pm10_latitude_n,pm10_longitude_n,cluster)

summ_plot_n <- summ_plot %>%
   dplyr::count(pm10_longitude_n, pm10_latitude_n, cluster)

colnames(summ_plot_n) <- c("pm10_longitude_n","pm10_latitude_n","cluster","count")

statesUSA <- map_data("state")

ggplot(data = statesUSA) +
  geom_polygon(aes(x = long, y = lat, group = group), fill = NA, color = "black",size = 0.15) +
  coord_map() +
  geom_point(data = summ_plot_n, aes(x = pm10_longitude_n, y = pm10_latitude_n, color = as.factor(cluste
  scale_x_continuous(limits = c(-125,-68)) +
  scale_y_continuous(limits = c(25,50))
```

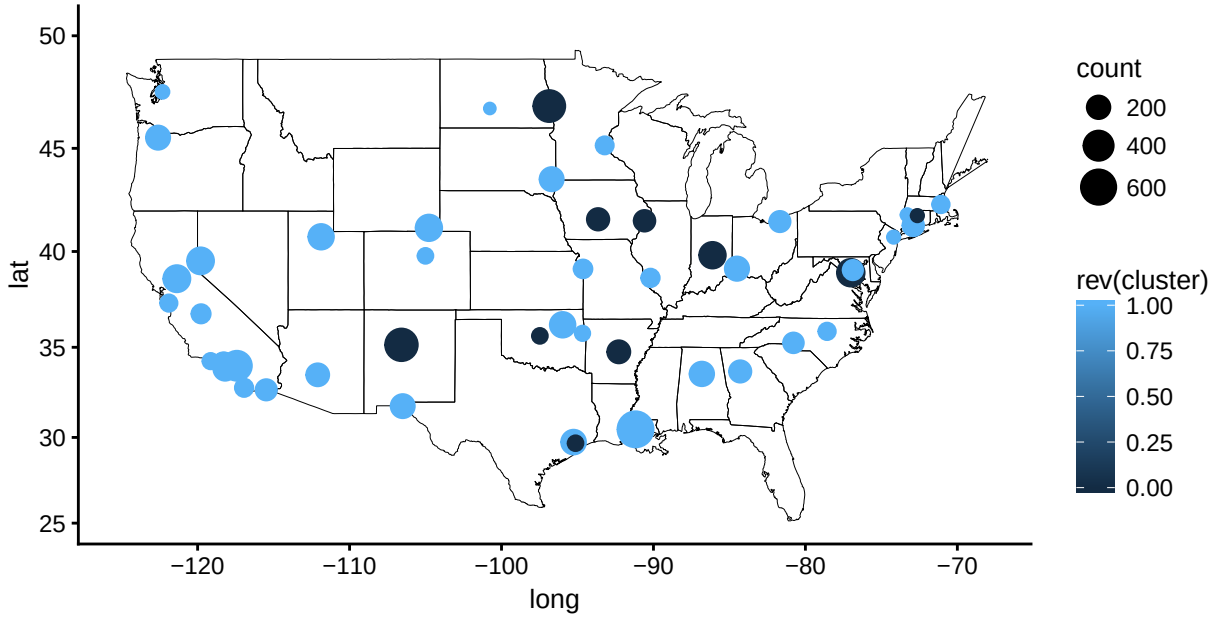


DBScan on Summer Data

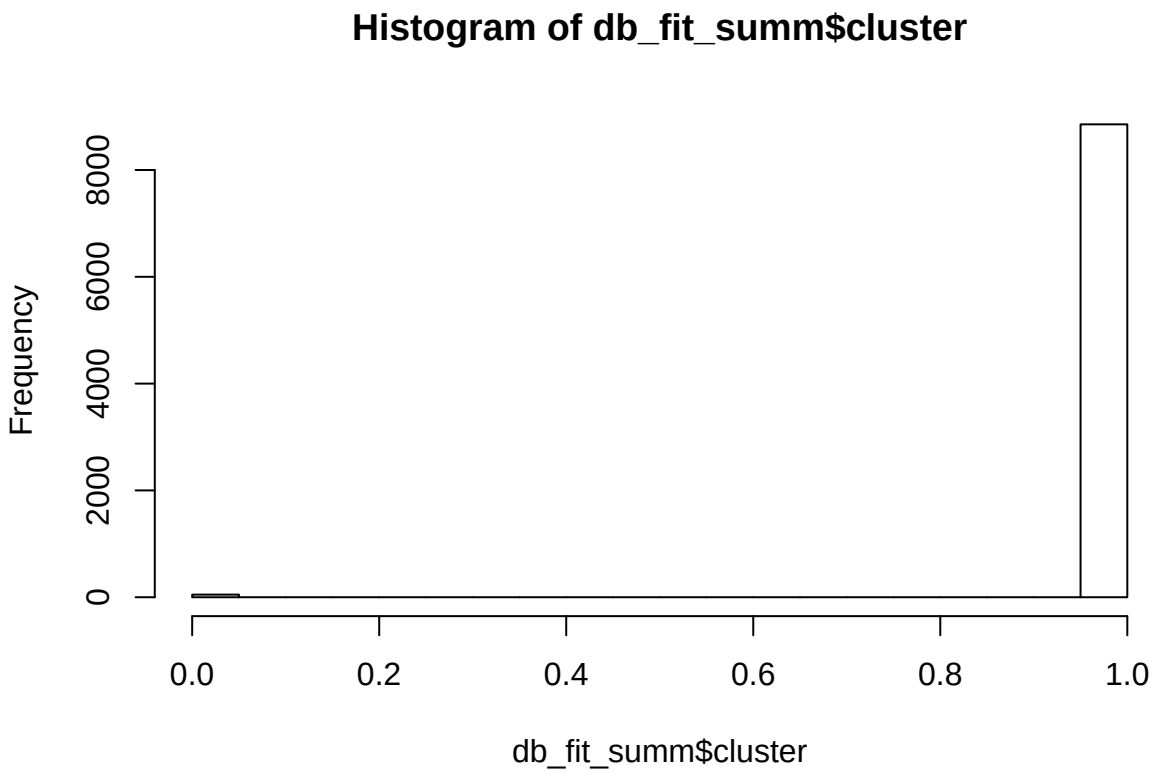
```
dbscan::kNNdistplot(scaled_summ_data, k = 100)
abline(h = 3, lty = 2)
```




```r
# DB Scan
set.seed(123)
db_fit_summ <- fpc::dbscan(scaled_summ_data, eps = 3 , MinPts = 20)


data_summer$db_cluster <- db_fit_summ$cluster

# Plotting on the geolocations
db_summ_plot <- data_summer %>%
  dplyr::select(pm10_latitude_n,pm10_longitude_n,db_cluster)

db_summ_plot_n <- db_summ_plot %>%
  dplyr::count(pm10_longitude_n, pm10_latitude_n, db_cluster)

colnames(db_summ_plot_n) <- c("pm10_longitude_n","pm10_latitude_n","cluster","count")

statesUSA <- map_data("state")

ggplot(data = statesUSA) +
  geom_polygon(aes(x = long, y = lat, group = group), fill = NA, color = "black", size =0.15) +
  coord_map() +
  geom_point(data = db_summ_plot_n, aes(x = pm10_longitude_n, y = pm10_latitude_n, color = rev(cluster)
  scale_x_continuous(limits = c(-125,-68)) +
  scale_y_continuous(limits = c(25,50))
```




```
data_summer$db_cluster <- NULL
hist(db_fit_summ$cluster)
```



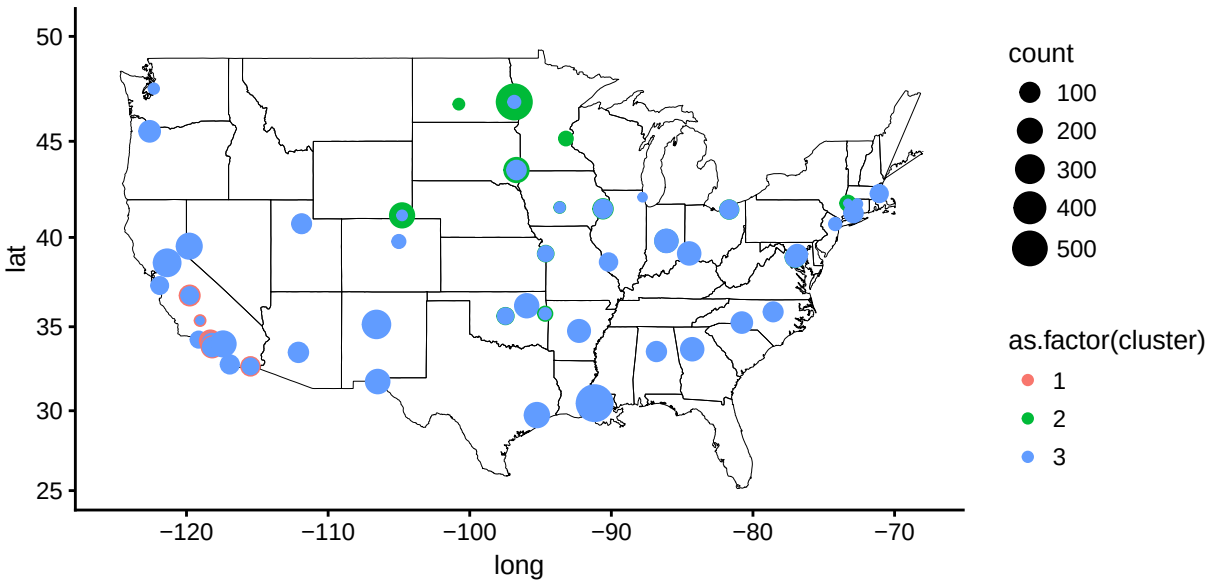
Winter Data

```
# Extracting the relevant variables and then scale them to have a mean of 0 and std with 1
data_win_n <- subset(data_winter, select = pm10_arithmetic_mean_n:wind_arithmetic_mean_n)
data_win_n <- subset(data_win_n, select = -no2_aqi_n)
```

```r
names(data_win_n)[1]<-"PM10"
names(data_win_n)[2]<-"CO"
names(data_win_n)[3]<-"SO2"
names(data_win_n)[4]<-"O3"
names(data_win_n)[5]<-"PM2.5"
names(data_win_n)[6]<-"TEMP"
names(data_win_n)[7]<-"RH"
names(data_win_n)[8]<-"NO2"
names(data_win_n)[9]<-"WIND"
scaled_win_data <- scale(data_win_n)
```

K-Means on Winter Data

```r
set.seed(123)
# Taking k as 3
fit_win <- kmeans(scaled_win_data, 3)

# Assigning cluster to the dataset
data_win_n$cluster <- fit_win$cluster
data_winter$cluster <- fit_win$cluster

# Plotting on the geolocations
win_plot <-data_winter %>%
  dplyr::select(pm10_latitude_n,pm10_longitude_n,cluster)

win_plot_n <- win_plot %>%
   dplyr::count(pm10_longitude_n, pm10_latitude_n, cluster)

colnames(win_plot_n) <- c("pm10_longitude_n","pm10_latitude_n","cluster","count")

ggplot(data = statesUSA) +
  geom_polygon(aes(x = long, y = lat, group = group), fill = NA, color = "black",size = 0.15) +
  coord_map() +
  geom_point(data = win_plot_n, aes(x = pm10_longitude_n, y = pm10_latitude_n, color = as.factor(cluste
  scale_x_continuous(limits = c(-125,-68)) +
  scale_y_continuous(limits = c(25,50))
```



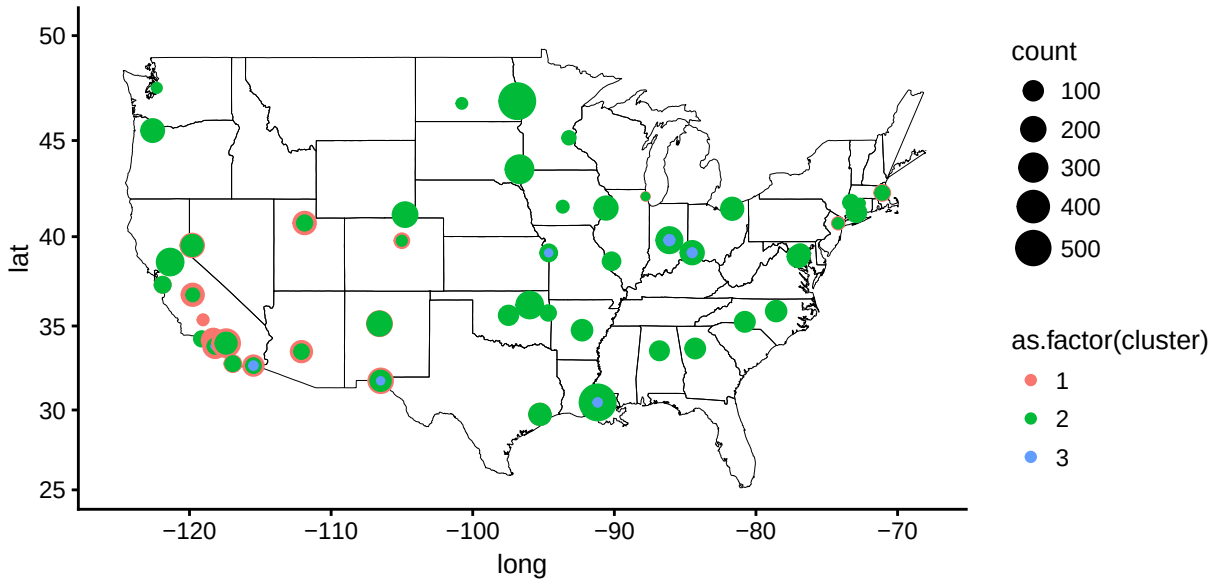

```
cew <- aggregate(data_win_n, by=list(cluster=fit_win$cluster), mean)
is.num <- sapply(cew, is.numeric)
cew[is.num] <- lapply(cew[is.num], round, 2)
cew[-11]
```

```
##   cluster  PM10   CO  SO2   O3 PM2.5  TEMP    RH   NO2 WIND
## 1       1 48.24 1.02 2.19 0.01 26.57 49.91 59.24 31.08 2.54
## 2       2 12.35 0.20 0.70 0.03  6.76 28.24 66.95  7.18 7.72
## 3       3 18.85 0.35 1.00 0.02  9.97 47.56 65.88 15.00 3.53
```

PAM on Winter Data

```
set.seed(123)
# Taking k as 3
fit_win <- clara(scaled_win_data, 3)

# Assigning cluster to the dataset
data_win_n$cluster <- fit_win$cluster
data_winter$cluster <- fit_win$cluster

# Plotting on the geolocations
win_plot <-data_winter %>%
  dplyr::select(pm10_latitude_n,pm10_longitude_n,cluster)

win_plot_n <- win_plot %>%
   dplyr::count(pm10_longitude_n, pm10_latitude_n, cluster)

colnames(win_plot_n) <- c("pm10_longitude_n","pm10_latitude_n","cluster","count")

ggplot(data = statesUSA) +
  geom_polygon(aes(x = long, y = lat, group = group), fill = NA, color = "black",size = 0.15) +
  coord_map() +
  geom_point(data = win_plot_n, aes(x = pm10_longitude_n, y = pm10_latitude_n, color = as.factor(cluste
```

```
  scale_x_continuous(limits = c(-125,-68)) +
  scale_y_continuous(limits = c(25,50))
```




```
t = data.frame(round(fit_win$medoids, 4))

t$PM10_n <- c((t$PM10*sd(data_winter$pm10_arithmetic_mean_n))+ mean(data_winter$pm10_arithmetic_mean_n))
t$CO_n <- c((t$CO*sd(data_winter$co_arithmetic_mean_n))+ mean(data_winter$co_arithmetic_mean_n))
t$SO2_n <- c((t$SO2*sd(data_winter$so2_arithmetic_mean_n))+ mean(data_winter$so2_arithmetic_mean_n))
t$O3_n <- c((t$O3*sd(data_winter$o3_arithmetic_mean_n))+ mean(data_winter$o3_arithmetic_mean_n))
t$PM2.5_n <- c((t$PM2.5*sd(data_winter$pm25f_arithmetic_mean_n))+ mean(data_winter$pm25f_arithmetic_mean
t$TEMP_n <- c((t$TEMP*sd(data_winter$temp_arithmetic_mean_n))+ mean(data_winter$temp_arithmetic_mean_n))
t$RH_n <- c((t$RH*sd(data_winter$rhdp_arithmetic_mean_n))+ mean(data_winter$rhdp_arithmetic_mean_n))
t$NO2_n <- c((t$NO2*sd(data_winter$no2_arithmetic_mean_n))+ mean(data_winter$no2_arithmetic_mean_n))
t$WIND_n <- c((t$WIND*sd(data_winter$wind_arithmetic_mean_n))+ mean(data_winter$wind_arithmetic_mean_n))

med_new <- t %>% dplyr::select(PM10_n:WIND_n)

med_new
```

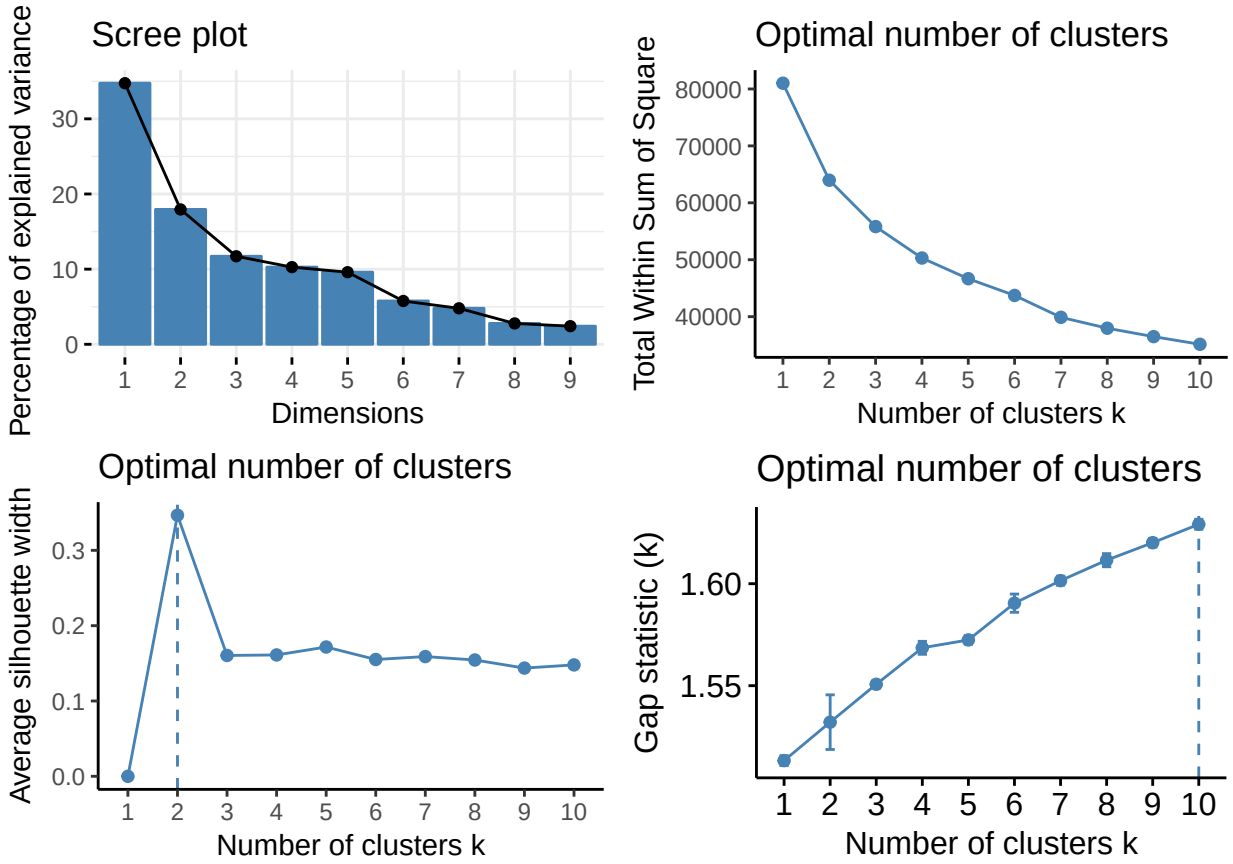
```
##         PM10_n      CO_n      SO2_n        O3_n  PM2.5_n   TEMP_n     RH_n
## 17312 26.99995 0.3208360  0.7763410 0.011708245 14.20049 41.50052 63.50055
## 29657 14.00009 0.1750062  0.2604343 0.021823530 10.79977 48.66624 79.08312
## 33519 80.00024 1.2541537 22.2770592 0.005916819 25.39993 58.29221 46.79155
##           NO2_n   WIND_n
## 17312 28.666381 2.450043
## 29657  9.549803 5.020715
## 33519 46.782883 1.966725
```

Fall Data

```
# Extracting the relevant variables and then scale them to have a mean of 0 and std with 1
data_fall_n <- subset(data_fall, select = pm10_arithmetic_mean_n:wind_arithmetic_mean_n)
data_fall_n <- subset(data_fall_n, select = -no2_aqi_n)
names(data_fall_n)[1]<-"PM10"
```

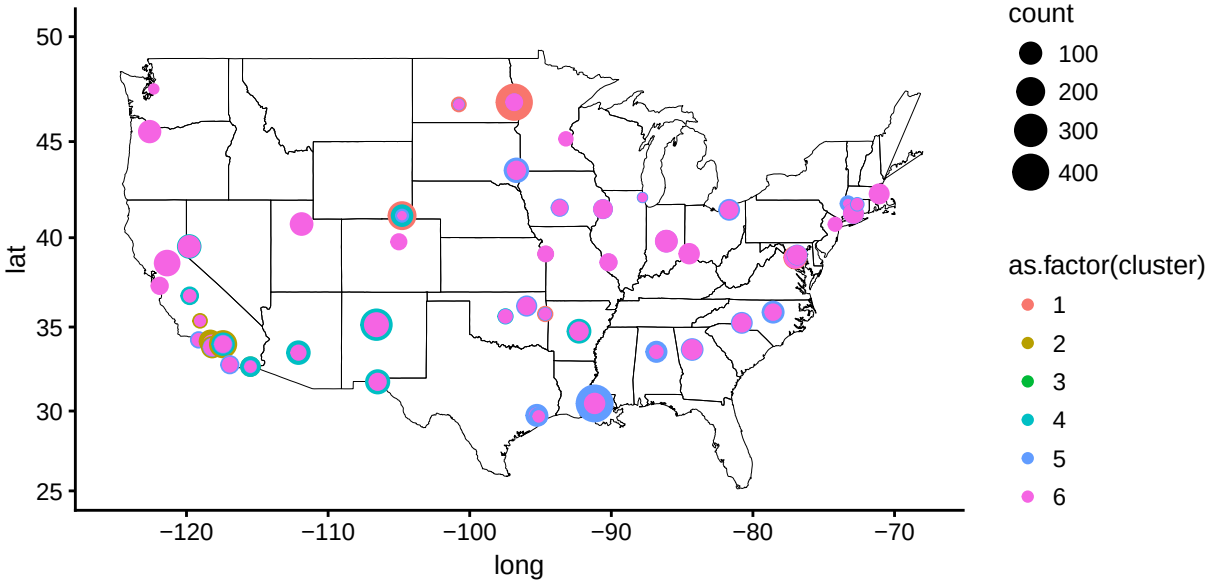
```r
names(data_fall_n)[2]<-"CO"
names(data_fall_n)[3]<-"SO2"
names(data_fall_n)[4]<-"O3"
names(data_fall_n)[5]<-"PM2.5"
names(data_fall_n)[6]<-"TEMP"
names(data_fall_n)[7]<-"RH"
names(data_fall_n)[8]<-"NO2"
names(data_fall_n)[9]<-"WIND"
scaled_fall_data <- scale(data_fall_n)
```

```r
set.seed(123)

theme_set(theme_cowplot(font_size=10)) # reduce default font size
fall_data.pca <- prcomp(scaled_fall_data,center = FALSE, scale = FALSE)
w1<- fviz_eig(fall_data.pca)
w2 <- fviz_nbclust(scaled_fall_data, kmeans, method = "wss") +
  theme_classic()
w3 <- fviz_nbclust(scaled_fall_data, kmeans, method = "silhouette") +
  theme_classic()
gap_statf <- clusGap(scaled_fall_data, FUN = kmeans, K.max = 10, B = 5)


# Plot gap statistic
w4 <- fviz_gap_stat(gap_statf)
plot_grid(w1,w2,w3,w4)
```



K-Means on Fall Data

```
set.seed(123)
# Taking k as 6
fit_fall <- kmeans(scaled_fall_data, 6)

# Assigning cluster to the dataset
data_fall_n$cluster <- fit_fall$cluster
data_fall$cluster <- fit_fall$cluster

# Plotting on the geolocations
fall_plot <-data_fall %>%
  dplyr::select(pm10_latitude_n,pm10_longitude_n,cluster)

fall_plot_n <- fall_plot %>%
   dplyr::count(pm10_longitude_n, pm10_latitude_n, cluster)

colnames(fall_plot_n) <- c("pm10_longitude_n","pm10_latitude_n","cluster","count")

statesUSA <- map_data("state")

ggplot(data = statesUSA) +
  geom_polygon(aes(x = long, y = lat, group = group), fill = NA, color = "black",size = 0.15) +
  coord_map() +
  geom_point(data = fall_plot_n, aes(x = pm10_longitude_n, y = pm10_latitude_n, color = as.factor(cluste
  scale_x_continuous(limits = c(-125,-68)) +
  scale_y_continuous(limits = c(25,50))
```



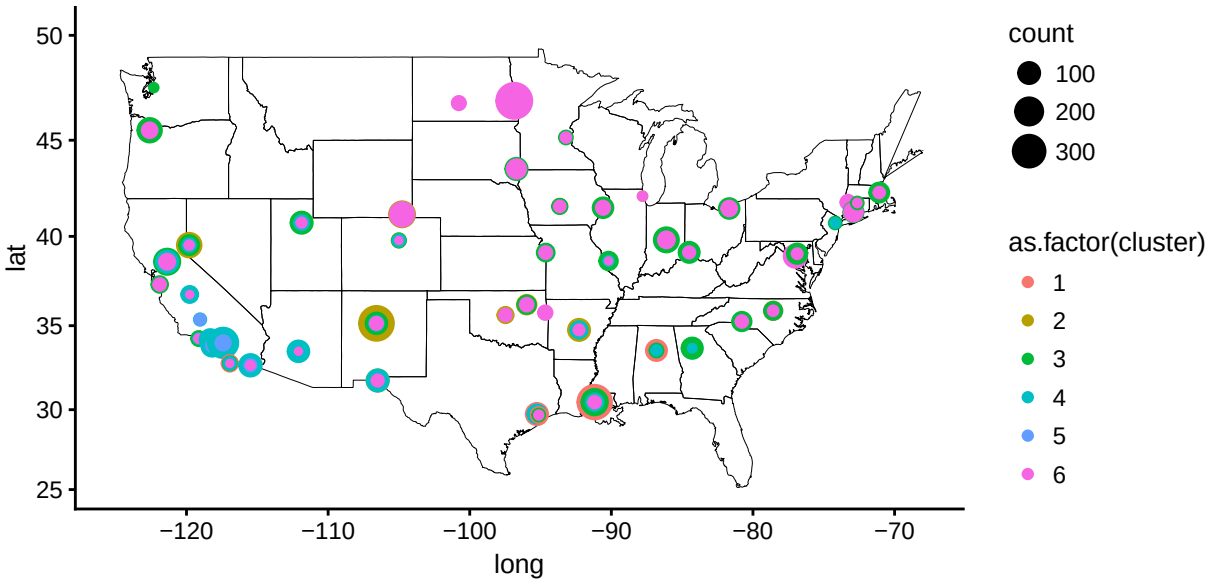

```
cef <- aggregate(data_fall_n, by=list(cluster=fit_fall$cluster), mean)
is.num <- sapply(cef, is.numeric)
cef[is.num] <- lapply(cef[is.num], round, 2)
cef[-11]
```

```
##   cluster  PM10   CO  SO2   O3 PM2.5  TEMP    RH   NO2 WIND
## 1       1 15.37 0.15 0.41 0.02  5.64 45.45 65.70  5.45 8.73
```

```
## 2       2 60.09 1.02 1.63 0.02 26.36 65.80 57.10 33.73 2.68
## 3       3 36.56 0.36 7.16 0.02 14.26 62.18 67.29 16.63 3.57
## 4       4 29.79 0.29 0.74 0.03  8.89 69.99 41.57 12.72 3.89
## 5       5 19.43 0.24 0.62 0.03  8.61 69.16 74.37  8.31 3.38
## 6       6 19.37 0.36 0.85 0.02  9.59 51.19 64.55 15.88 2.83
```

Partition Around Medoids for Fall Data

```
set.seed(123)
# Taking k as 6
fit_fall <- clara(scaled_fall_data, 6)

# Assigning cluster to the dataset
data_fall_n$cluster <- fit_fall$cluster
data_fall$cluster <- fit_fall$cluster

# Plotting on the geolocations
fall_plot <-data_fall %>%
  dplyr::select(pm10_latitude_n,pm10_longitude_n,cluster)

fall_plot_n <- fall_plot %>%
   dplyr::count(pm10_longitude_n, pm10_latitude_n, cluster)

colnames(fall_plot_n) <- c("pm10_longitude_n","pm10_latitude_n","cluster","count")

statesUSA <- map_data("state")

ggplot(data = statesUSA) +
  geom_polygon(aes(x = long, y = lat, group = group), fill = NA, color = "black",size = 0.15) +
  coord_map() +
  geom_point(data = fall_plot_n, aes(x = pm10_longitude_n, y = pm10_latitude_n, color = as.factor(cluste
  scale_x_continuous(limits = c(-125,-68)) +
  scale_y_continuous(limits = c(25,50))
```



```
t = data.frame(round(fit_fall$medoids, 4))

t$PM10_n <- c((t$PM10*sd(data_fall$pm10_arithmetic_mean_n))+ mean(data_fall$pm10_arithmetic_mean_n))
t$CO_n <- c((t$CO*sd(data_fall$co_arithmetic_mean_n))+ mean(data_fall$co_arithmetic_mean_n))
t$SO2_n <- c((t$SO2*sd(data_fall$so2_arithmetic_mean_n))+ mean(data_fall$so2_arithmetic_mean_n))
t$O3_n <- c((t$O3*sd(data_fall$o3_arithmetic_mean_n))+ mean(data_fall$o3_arithmetic_mean_n))
t$PM2.5_n <- c((t$PM2.5*sd(data_fall$pm25f_arithmetic_mean_n))+ mean(data_fall$pm25f_arithmetic_mean_n))
t$TEMP_n <- c((t$TEMP*sd(data_fall$temp_arithmetic_mean_n))+ mean(data_fall$temp_arithmetic_mean_n))
t$RH_n <- c((t$RH*sd(data_fall$rhdp_arithmetic_mean_n))+ mean(data_fall$rhdp_arithmetic_mean_n))
t$NO2_n <- c((t$NO2*sd(data_fall$no2_arithmetic_mean_n))+ mean(data_fall$no2_arithmetic_mean_n))
t$WIND_n <- c((t$WIND*sd(data_fall$wind_arithmetic_mean_n))+ mean(data_fall$wind_arithmetic_mean_n))

med_new <- t %>% dplyr::select(PM10_n:WIND_n)

med_new
```

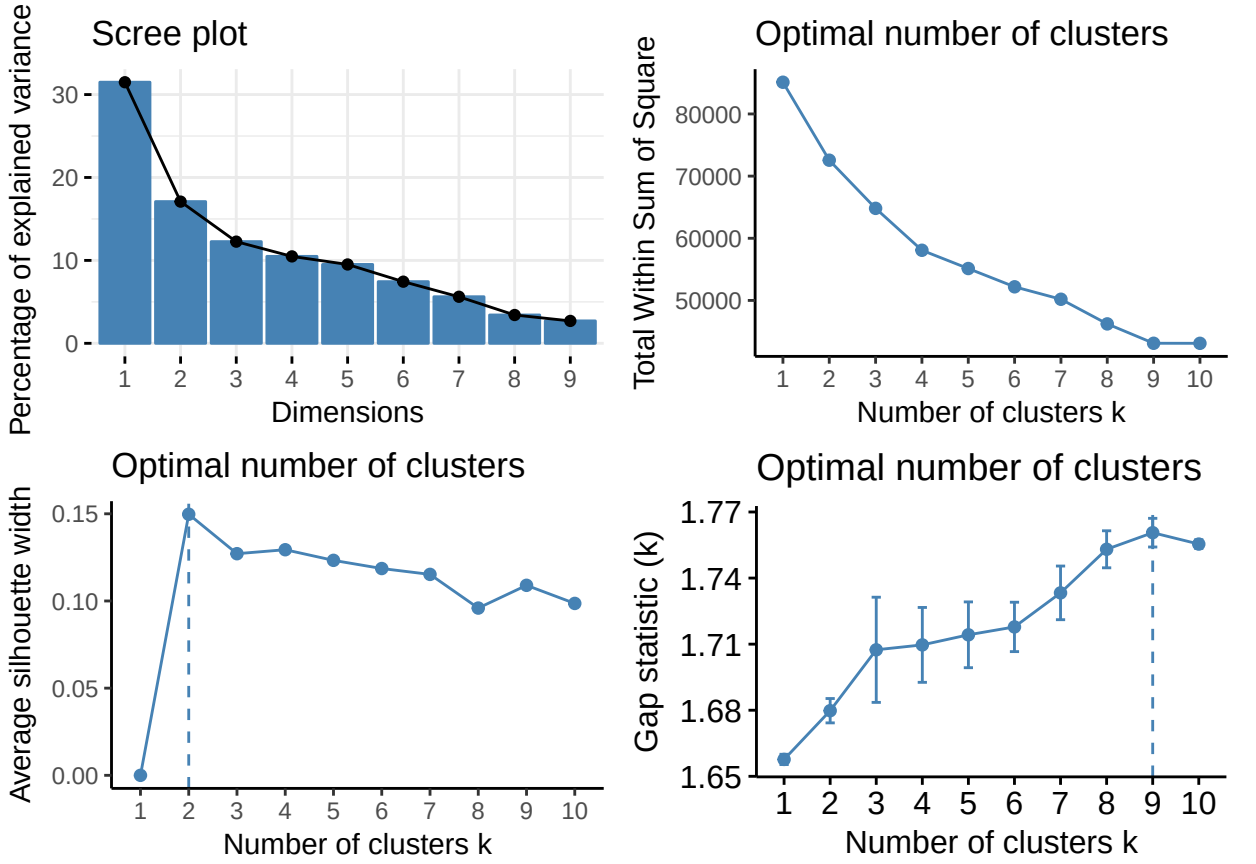
```
##           PM10_n      CO_n       SO2_n       O3_n   PM2.5_n   TEMP_n
## 18500  25.000309 0.1875056 0.518681201 0.02729435  9.949933 83.45770
## 23588  13.000987 0.1541735 0.374974946 0.03458762  4.899953 69.77040
## 27711  19.999409 0.2250080 1.443014030 0.01647108  7.100342 49.40865
## 7438   50.000754 0.6958274 1.575779686 0.02791666 16.600345 71.41716
## 6516  122.000740 2.0541624 5.188690968 0.02574985 47.699621 73.33322
## 15661   7.000312 0.1749948 0.005064399 0.01945816  1.800038 52.41737
##           RH_n     NO2_n   WIND_n
## 18500 70.91716  6.670856 2.812461
## 23588 50.16607  8.257946 4.433399
## 27711 80.16716 11.565281 2.937502
## 7438  39.33286 26.564798 2.245862
## 6516  59.13886 41.363604 3.400125
## 15661 73.16641  2.749637 7.483258
```

Spring Data

```
# Extracting the relevant variables and then scale them to have a mean of 0 and std with 1
data_spr_n <- subset(data_spring, select = pm10_arithmetic_mean_n:wind_arithmetic_mean_n)
data_spr_n <- subset(data_spr_n, select = -no2_aqi_n)
names(data_spr_n)[1]<-"PM10"
names(data_spr_n)[2]<-"CO"
names(data_spr_n)[3]<-"SO2"
names(data_spr_n)[4]<-"O3"
names(data_spr_n)[5]<-"PM2.5"
names(data_spr_n)[6]<-"TEMP"
names(data_spr_n)[7]<-"RH"
names(data_spr_n)[8]<-"NO2"
names(data_spr_n)[9]<-"WIND"
scaled_spr_data <- scale(data_spr_n)
```

```
set.seed(123)

theme_set(theme_cowplot(font_size=10)) # reduce default font size
spr_data.pca <- prcomp(scaled_spr_data,center = FALSE, scale = FALSE)
w1<- fviz_eig(spr_data.pca)
w2 <- fviz_nbclust(scaled_spr_data, clara, method = "wss") +
```

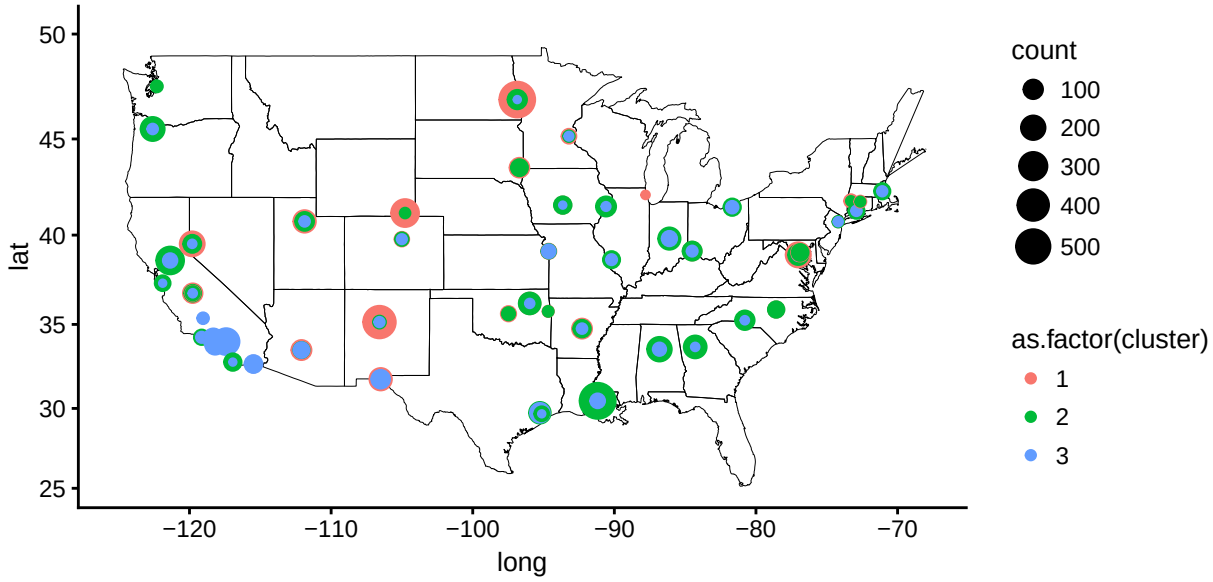
```
  theme_classic()
w3 <- fviz_nbclust(scaled_spr_data, clara, method = "silhouette") +
  theme_classic()
gap_statsp <- clusGap(scaled_spr_data, FUN = clara, K.max = 10, B = 5)

# Plot gap statistic
w4 <- fviz_gap_stat(gap_statsp)
plot_grid(w1,w2,w3,w4)
```



K-Means for Spring Data

```
set.seed(123)
# Taking k as 3
fit_spr <- kmeans(scaled_spr_data, 3)

# Assigning cluster to the dataset
data_spr_n$cluster <- fit_spr$cluster
data_spring$cluster <- fit_spr$cluster

# Plotting on the geolocations
spr_plot <-data_spring %>%
  dplyr::select(pm10_latitude_n,pm10_longitude_n,cluster)
```

```
spr_plot_n <- spr_plot %>%
   dplyr::count(pm10_longitude_n, pm10_latitude_n, cluster)

colnames(spr_plot_n) <- c("pm10_longitude_n","pm10_latitude_n","cluster","count")

statesUSA <- map_data("state")

ggplot(data = statesUSA) +
  geom_polygon(aes(x = long, y = lat, group = group), fill = NA, color = "black",size = 0.15) +
  coord_map() +
  geom_point(data = spr_plot_n, aes(x = pm10_longitude_n, y = pm10_latitude_n, color = as.factor(cluste
  scale_x_continuous(limits = c(-125,-68)) +
  scale_y_continuous(limits = c(25,50))
```



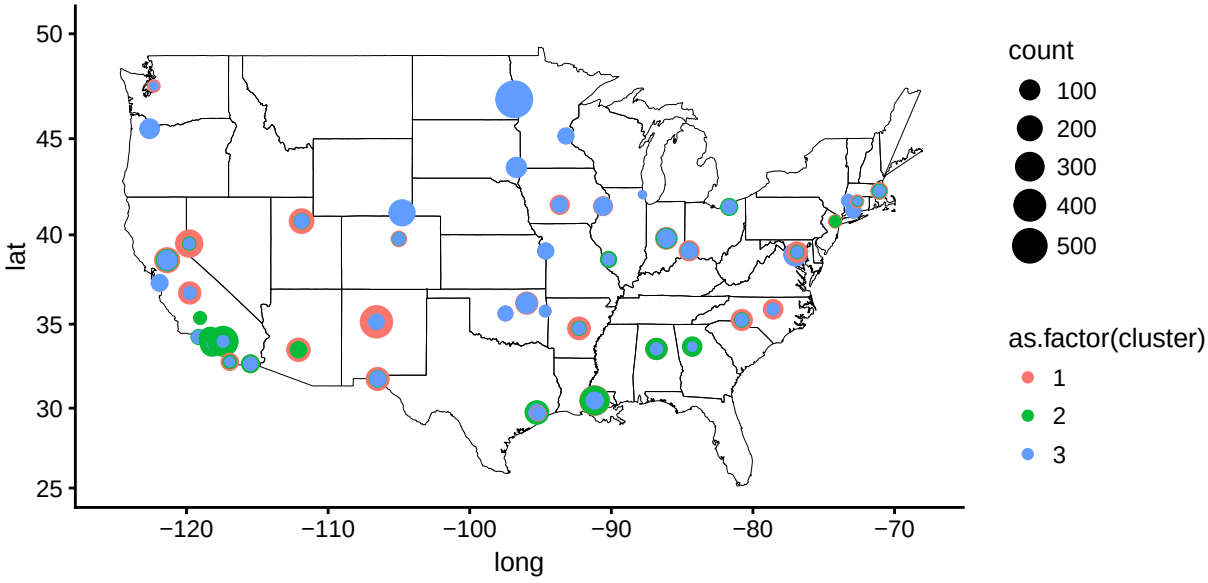

```
data_spr_n$cluster <- NULL
data_spring$cluster <- NULL

ces <- aggregate(data_spr_n, by=list(cluster=fit_spr$cluster), mean)
is.num <- sapply(ces, is.numeric)
ces[is.num] <- lapply(ces[is.num], round, 2)
ces[-11]
```

```
##   cluster  PM10   CO  SO2   O3 PM2.5  TEMP    RH   NO2 WIND
## 1       1 18.87 0.21 0.50 0.04  6.56 54.90 47.61  7.50 6.58
## 2       2 18.71 0.27 0.67 0.03  8.49 60.76 69.80  9.86 3.72
## 3       3 49.12 0.59 2.15 0.03 17.72 64.75 53.06 23.66 3.96
```

Partition Around Medoids for K-Means

```
set.seed(123)

# Taking k as 3
fit_spr <- clara(scaled_spr_data, 3)
```

```
# Assigning cluster to the dataset
data_spr_n$cluster <- fit_spr$cluster
data_spring$cluster <- fit_spr$cluster

# Plotting on the geolocations
spr_plot <-data_spring %>%
  dplyr::select(pm10_latitude_n,pm10_longitude_n,cluster)

spr_plot_n <- spr_plot %>%
   dplyr::count(pm10_longitude_n, pm10_latitude_n, cluster)

colnames(spr_plot_n) <- c("pm10_longitude_n","pm10_latitude_n","cluster","count")

statesUSA <- map_data("state")

ggplot(data = statesUSA) +
  geom_polygon(aes(x = long, y = lat, group = group), fill = NA, color = "black",size = 0.15) +
  coord_map() +
  geom_point(data = spr_plot_n, aes(x = pm10_longitude_n, y = pm10_latitude_n, color = as.factor(cluste
  scale_x_continuous(limits = c(-125,-68)) +
  scale_y_continuous(limits = c(25,50))
```




```
t = data.frame(round(fit_spr$medoids, 4))

t$PM10_n <- c((t$PM10*sd(data_spring$pm10_arithmetic_mean_n))+ mean(data_spring$pm10_arithmetic_mean_n)
t$CO_n <- c((t$CO*sd(data_spring$co_arithmetic_mean_n))+ mean(data_spring$co_arithmetic_mean_n))
t$SO2_n <- c((t$SO2*sd(data_spring$so2_arithmetic_mean_n))+ mean(data_spring$so2_arithmetic_mean_n))
t$O3_n <- c((t$O3*sd(data_spring$o3_arithmetic_mean_n))+ mean(data_spring$o3_arithmetic_mean_n))
t$PM2.5_n <- c((t$PM2.5*sd(data_spring$pm25f_arithmetic_mean_n))+ mean(data_spring$pm25f_arithmetic_mea
t$TEMP_n <- c((t$TEMP*sd(data_spring$temp_arithmetic_mean_n))+ mean(data_spring$temp_arithmetic_mean_n)
t$RH_n <- c((t$RH*sd(data_spring$rhdp_arithmetic_mean_n))+ mean(data_spring$rhdp_arithmetic_mean_n))
t$NO2_n <- c((t$NO2*sd(data_spring$no2_arithmetic_mean_n))+ mean(data_spring$no2_arithmetic_mean_n))
t$WIND_n <- c((t$WIND*sd(data_spring$wind_arithmetic_mean_n))+ mean(data_spring$wind_arithmetic_mean_n)
```

```r
med_new <- t %>% dplyr::select(PM10_n:WIND_n)

med_new
```

```
##         PM10_n      CO_n     SO2_n       O3_n   PM2.5_n   TEMP_n     RH_n
## 17039 10.99930 0.2333356 0.2437932 0.03994100  6.999886 59.95815 53.74985
## 24312 23.00072 0.3479109 1.2645763 0.02849999 16.499913 58.99996 72.79253
## 27231 14.99977 0.1999965 0.1083061 0.03570633  8.799844 49.04157 63.20758
##           NO2_n   WIND_n
## 17039  8.620851 4.587415
## 24312 16.591013 1.566611
## 27231  6.171068 7.849883
```