#Import dataset

```
BOROUGH <- readxl::read_excel("/Users/shimonyagrawal/Desktop/NYC Real Estate/BOROUGH.xlsx")
NEIGHBORHOOD <- readxl::read_excel("/Users/shimonyagrawal/Desktop/NYC Real Estate/NEIGHBORHOOD.xlsx")
BUILDING_CLASS <- readxl::read_excel("/Users/shimonyagrawal/Desktop/NYC Real Estate/BUILDING_CLASS.xlsx
NYC_HISTORICAL <- readxl::read_excel("/Users/shimonyagrawal/Desktop/NYC Real Estate/NYC_HISTORICAL.xlsx
```

#Install packages for analysis

```
tinytex::install_tinytex()
```

```
## Warning: Detected an existing tlmgr at /usr/local/bin/tlmgr. It seems TeX
## Live has been installed (check tinytex::tinytex_root()). You are recommended
## to uninstall it, although TinyTeX should work well alongside another LaTeX
## distribution if a LaTeX document is compiled through tinytex::latexmk().

## TinyTeX installed to /Users/shimonyagrawal/Library/TinyTeX
```

```
install.packages("readxl")
```

```
##
## The downloaded binary packages are in
##  /var/folders/rj/t11km2gs693dyq4szgcrm4vr0000gn/T//RtmplaxW4t/downloaded_packages
```

```
install.packages("DBI")
```

```
##
## The downloaded binary packages are in
##  /var/folders/rj/t11km2gs693dyq4szgcrm4vr0000gn/T//RtmplaxW4t/downloaded_packages
```

```
install.packages("odbc")
```

```
##
## The downloaded binary packages are in
##  /var/folders/rj/t11km2gs693dyq4szgcrm4vr0000gn/T//RtmplaxW4t/downloaded_packages
```

```
install.packages("tidyverse")
```

```
##
## The downloaded binary packages are in
##  /var/folders/rj/t11km2gs693dyq4szgcrm4vr0000gn/T//RtmplaxW4t/downloaded_packages
```

```
install.packages("lubridate")
```

```
##
## The downloaded binary packages are in
##  /var/folders/rj/t11km2gs693dyq4szgcrm4vr0000gn/T//RtmplaxW4t/downloaded_packages
```

```r
install.packages("cluster")
```

```
##
## The downloaded binary packages are in
##  /var/folders/rj/t11km2gs693dyq4szgcrm4vr0000gn/T//RtmplaxW4t/downloaded_packages
```

```r
install.packages("factoextra")
```

```
##
## The downloaded binary packages are in
##  /var/folders/rj/t11km2gs693dyq4szgcrm4vr0000gn/T//RtmplaxW4t/downloaded_packages
```

```r
library(readxl)
library(DBI)
library(odbc)
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------- tidyver
```

```
## v ggplot2 3.3.0     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   0.8.5
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------------------------------------- tidyverse_con
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:dplyr':
##
##     intersect, setdiff, union
```

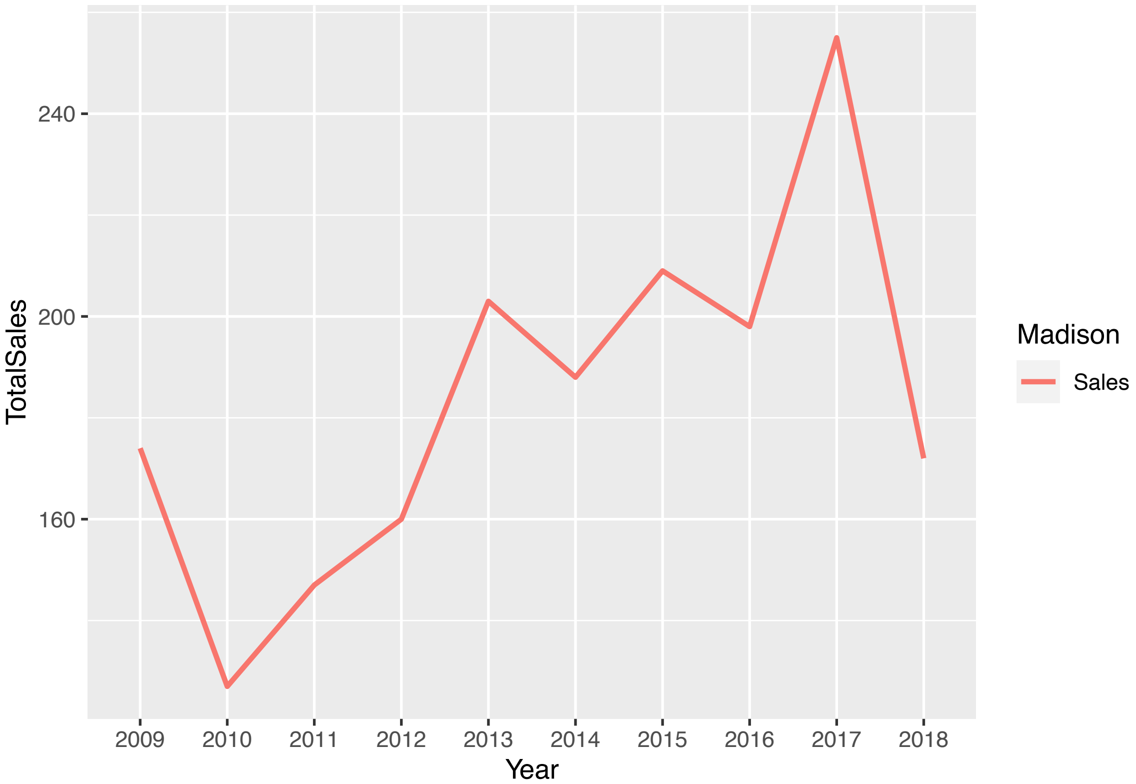
```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
#Descriptive Statistics for Neighborhood Madison (149)
#create dataframe with required data and filter N/A or missing values

NYCdf <- NYC_HISTORICAL %>%
  left_join(NEIGHBORHOOD, by= "NEIGHBORHOOD_ID") %>%
  left_join(BUILDING_CLASS, by= c("BUILDING_CLASS_FINAL_ROLL"="BUILDING_CODE_ID")) %>%
  select (NEIGHBORHOOD_ID, NEIGHBORHOOD_NAME, SALE_DATE, SALE_PRICE, GROSS_SQUARE_FEET, RESIDENTIAL_UNI
  filter(SALE_PRICE >0) %>%
  separate(SALE_DATE, c("Year", "Month", "Date"))%>%
  group_by(Year)

view(NYCdf)
```

```
#Q: Provide descriptive statistics for real estate sales in your neighborhood
#Total Sales since 2009 (Madison)

Madison <- filter(NYCdf,NEIGHBORHOOD_ID=="149", Year> 2008) %>%
  group_by(Year)
Madison.Sales <- summarise(Madison, TotalSales = n())%>%
  filter (TotalSales > 0)
view(Madison.Sales)

ggplot()+
  geom_line(data=Madison.Sales, size=1, aes(x=Year, y=TotalSales, group=1, color="blue"))+
  scale_color_discrete(name="Madison", label="Sales")
```



```r
#Mean sales and mean gross square feet since 2009 (Madison)

Madison.SalesSqFt <- filter(NYCdf,SALE_PRICE>0, GROSS_SQUARE_FEET>0, TYPE=="RESIDENTIAL", NEIGHBORHOOD_I
  summarise(TotalSalePrice = sum(SALE_PRICE), TotalGrossFeet= sum(GROSS_SQUARE_FEET))
Madison.MeanSaleSqFt <- summarise (Madison.SalesSqFt, MeanSales= mean(TotalSalePrice), MeanGrossFeet = r
view(Madison.SalesSqFt)
view(Madison.MeanSaleSqFt)

#Five number summary for sales and gross square feet
summary(Madison.SalesSqFt)
```

```
##      Year           TotalSalePrice      TotalGrossFeet
##  Length:16          Min.   : 59983969   Min.   :207920
##  Class :character   1st Qu.: 79026628   1st Qu.:256902
##  Mode  :character   Median : 94715399   Median :272643
##                     Mean   : 95795337   Mean   :309978
##                     3rd Qu.:117604017   3rd Qu.:358774
##                     Max.   :129741460   Max.   :506419
```

```r
#Proportion of Residential, Commercial, mixed and other sales

Madison.Proportion <- NYCdf %>%
  select(NEIGHBORHOOD_NAME, NEIGHBORHOOD_ID, TYPE, RESIDENTIAL_UNITS, COMMERCIAL_UNITS)
```

```
## Adding missing grouping variables: 'Year'
```

```r
Madison.Proportion <- group_by(NYCdf, TYPE) %>%
  drop_na() %>%
  summarise(Total_Units = (sum(RESIDENTIAL_UNITS, na.rm = T) + sum(COMMERCIAL_UNITS, na.rm = T))) %>%
  mutate(Proportion = Total_Units/sum(Total_Units))
view (Madison.Proportion)
```

```r
#Standard Deviation for Sales Prices in Residential properties

SdSalePrices.Residential <- filter(NYCdf, TYPE=="RESIDENTIAL", SALE_PRICE>0,
                                   NEIGHBORHOOD_ID==149, Year>2008) %>%
  group_by(Year) %>%
  select(SALE_PRICE, TYPE,RESIDENTIAL_UNITS) %>%
  summarise(TotalSales = sum(SALE_PRICE, na.rm = T),
            TotalResidentialUnits = sum(RESIDENTIAL_UNITS, na.rm = T)) %>%
  summarise(SdSalePrice= sd(TotalSales))
```

```
## Adding missing grouping variables: 'Year'
```

```r
view(SdSalePrices.Residential)

#Correlation between sale price and gross square feet in residential units
cor(Madison.SalesSqFt[c(2,3)])
```

```
##                TotalSalePrice TotalGrossFeet
## TotalSalePrice      1.0000000      0.5038537
## TotalGrossFeet      0.5038537      1.0000000
```

#Q: Perform K-means clustering, comparing your neighborhood to other neighborhoods

```
#Comparing K-means cluster for Median Sales for Residential properties of all neighborhoods

KPI1 <- filter(NYCdf, TYPE=="RESIDENTIAL", SALE_PRICE>0,Year>2008)%>%
  group_by(NEIGHBORHOOD_ID)%>%
  select(SALE_PRICE, TYPE, RESIDENTIAL_UNITS) %>%
  summarise(TotalSales = sum(SALE_PRICE, na.rm = T),
         TotalResidentialUnits = sum(RESIDENTIAL_UNITS, na.rm = T),
         MedSales=median(TotalSales))
```

```
## Adding missing grouping variables: 'NEIGHBORHOOD_ID'
```

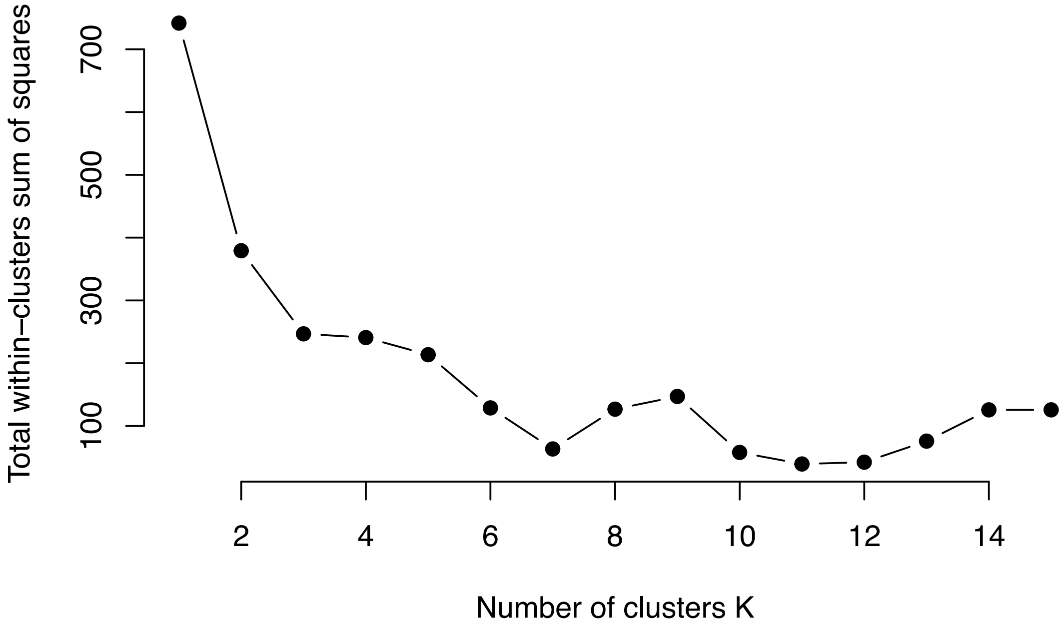
```
view(KPI1)

zscores1 <- select(KPI1, "MedSales", "TotalResidentialUnits")
zscores1 <- as.data.frame(scale(zscores1))

clustering1 <- kmeans(zscores1, centers = 5)
clustering1[["centers"]]
```

```
##     MedSales TotalResidentialUnits
## 1 -0.3905409            -0.5947271
## 2  0.1797548             0.1585192
## 3  0.2872743             1.4722689
## 4  6.2641644             0.1539046
## 5  1.5720662             4.1504229
```

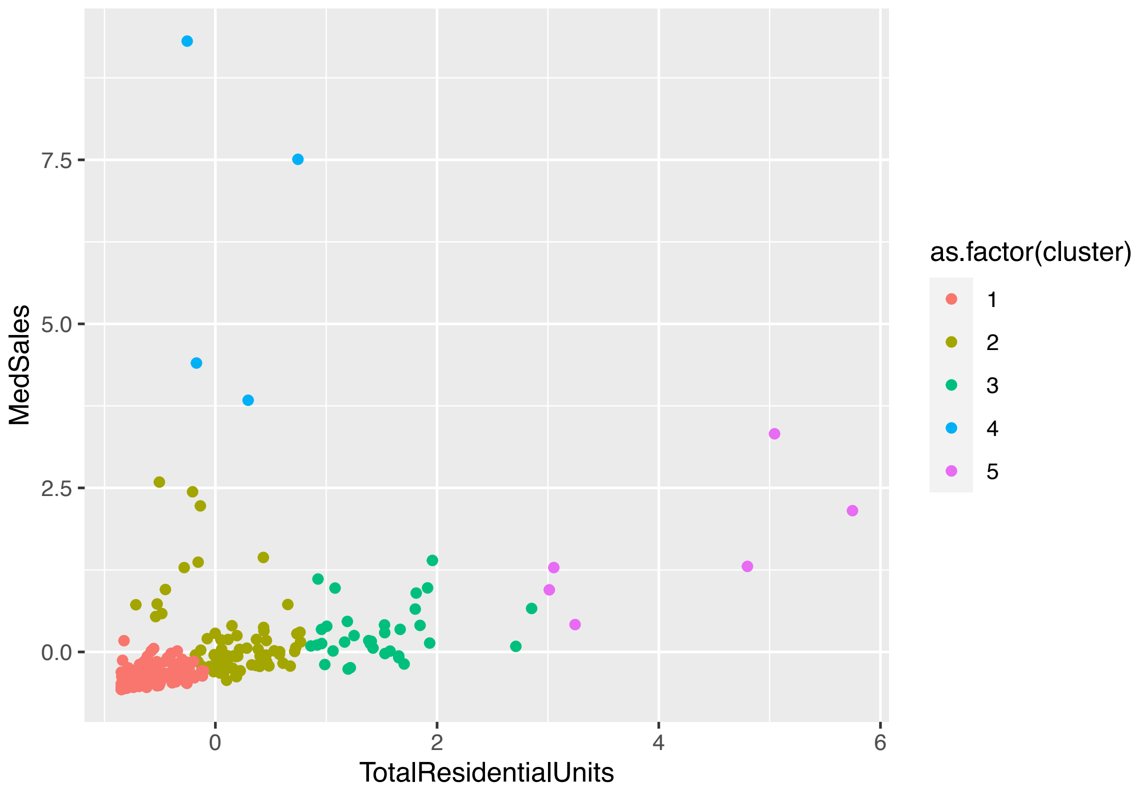
```
zscores1 <- cbind(zscores1, cluster=clustering1$cluster)
zScores1 <- cbind(KPI1[c(1)], zscores1)

#finding the optimal number of clusters using the elbow method
k.max <- 15
data <- zscores1
wss <- sapply(1:k.max,
              function(k){kmeans(data, k)$tot.withinss})
wss
```

```
##  [1] 741.66284 379.12856 246.77906 240.83989 213.54133 128.82846  63.43015
##  [8] 126.80660 147.07042  57.92563  39.49458  42.35418  75.89634 125.66917
## [15] 125.70310
```

```
plot(1:k.max, wss,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")
```




```
ggplot(zScores1)+
  geom_point (aes(x=TotalResidentialUnits, y=MedSales, color=as.factor(cluster)))
```




```r
#Comparing K-means cluster for Number of Sales for Residential properties of  all neighborhoods

KPI2 <- filter(NYCdf, SALE_PRICE>0,Year>2008)%>%
  group_by(NEIGHBORHOOD_ID)%>%
  select(SALE_PRICE, TYPE, RESIDENTIAL_UNITS, COMMERCIAL_UNITS)%>%
  drop_na()%>%
  summarise(TotalUnits=sum(RESIDENTIAL_UNITS+COMMERCIAL_UNITS), TotalResidentialUnits = sum(RESIDENTIAL_
```

```
## Adding missing grouping variables: `NEIGHBORHOOD_ID`
```

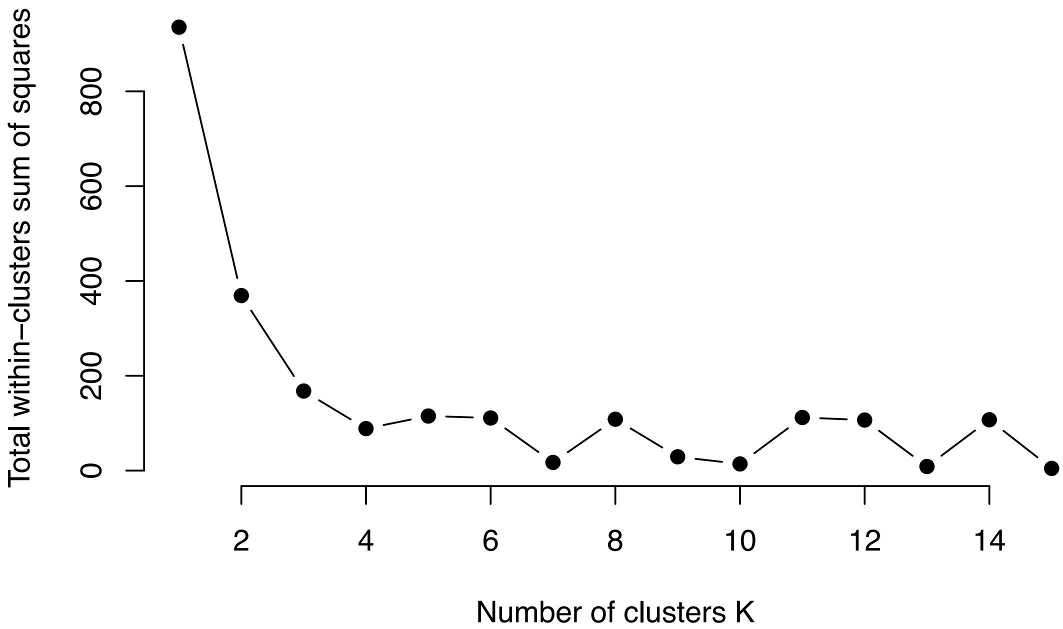
```r
zscores2 <- select(KPI2, "TotalResidentialUnits" , "TotalUnits")
zscores2 <- scale(zscores2)
view(zscores2)

clustering2 <- kmeans (zscores2, centers = 5)
clustering2[["centers"]]
```

```
##   TotalResidentialUnits TotalUnits
## 1             1.4234288  1.4234288
## 2             3.8167084  3.8167084
## 3            -0.7070348 -0.7070348
## 4             0.3857631  0.3857631
## 5            -0.2413817 -0.2413817
```

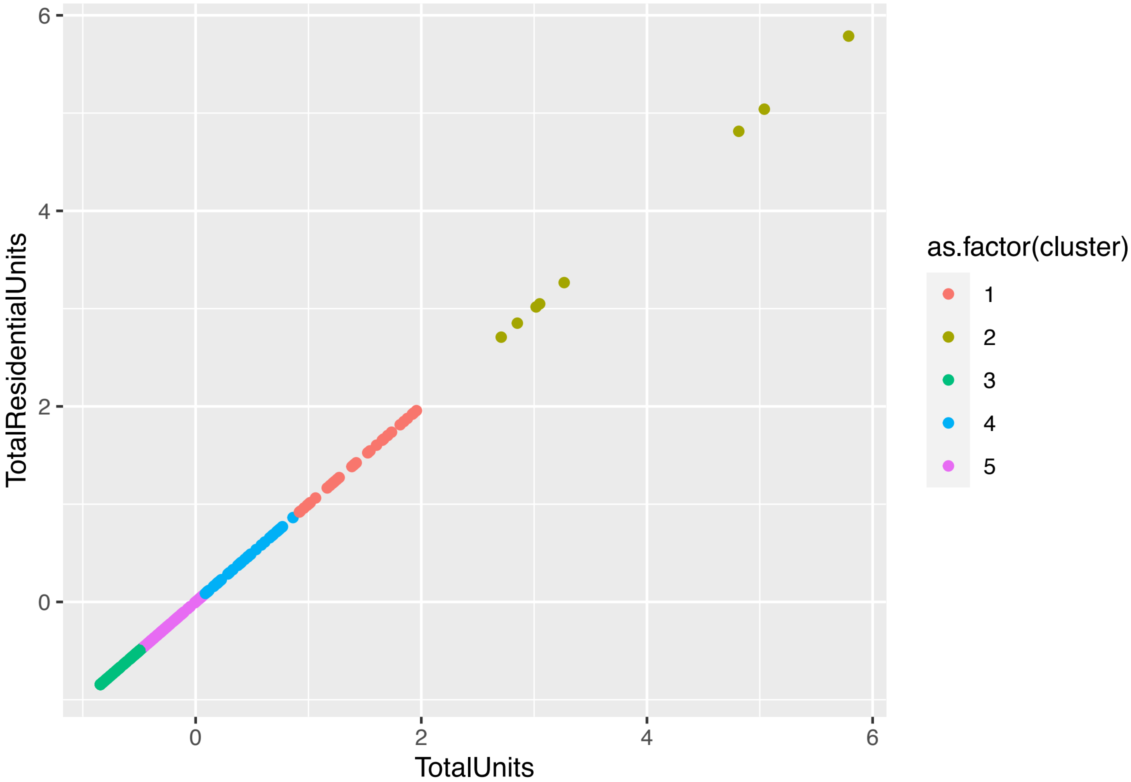
```r
zscores2 <- cbind(zscores2, cluster=clustering2$cluster)
zScores2 <- cbind.data.frame(KPI2[c(1)], zscores2)


#finding the optimal number of clusters using the elbow method
k.max <- 15
data <- zscores2
wss <- sapply(1:k.max,
              function(k){kmeans(data, k)$tot.withinss})

wss
```

```
##  [1] 935.359848 369.199568 167.791957  88.587275 115.091606 110.960946
##  [7]  17.169765 108.311993  29.130218  14.056056 111.911075 106.601869
## [13]   8.577114 107.233681   4.558423
```

```r
plot(1:k.max,
     wss,type="b",pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")
```




```r
ggplot(zScores2)+
  geom_point(aes(x=TotalUnits, y=TotalResidentialUnits, color=as.factor(cluster)))
```




```
#Comparing K-means cluster for Proportion of Residential Sales for all neighborhoods

KPI3 <- filter (NYCdf,Year>2008,SALE_PRICE>0, TYPE=="RESIDENTIAL")%>%
  group_by(NEIGHBORHOOD_ID)%>%
  select(RESIDENTIAL_UNITS, NEIGHBORHOOD_ID) %>%
  summarise(TotalUnits=n(), TotalResUnits = sum(RESIDENTIAL_UNITS))%>%
  drop_na()%>%
  mutate(Proportion=TotalUnits/sum(TotalUnits))

zscores3 <- select(KPI3,"TotalResUnits", "Proportion")
zscores3 <- scale(zscores3)
view(zscores3)

clustering3 <- kmeans (zscores3, centers = 5)
clustering3[["centers"]]
```

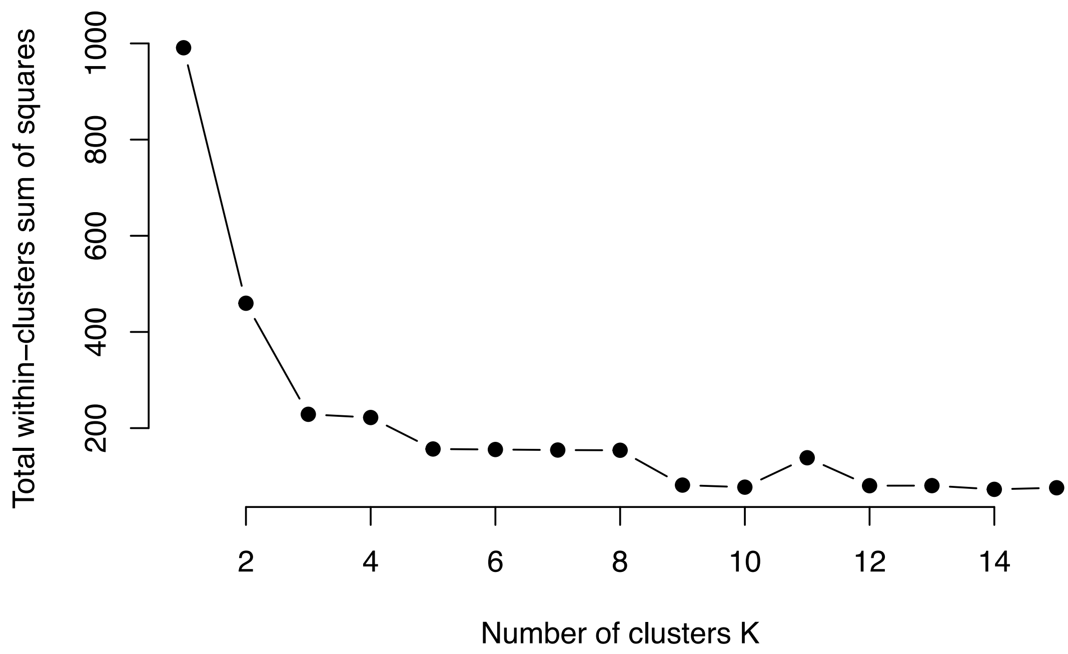
```
##   TotalResUnits Proportion
## 1    4.70925712  2.9089869
## 2   -0.62106005 -0.6145238
## 3    1.47407328  0.6920521
## 4    0.68387739  3.0427764
## 5    0.04098084  0.1789817
```

```
zscores3 <- cbind(zscores3, cluster=clustering3$cluster)
zScores3 <- cbind.data.frame(KPI3[c(1)], zscores3)
```

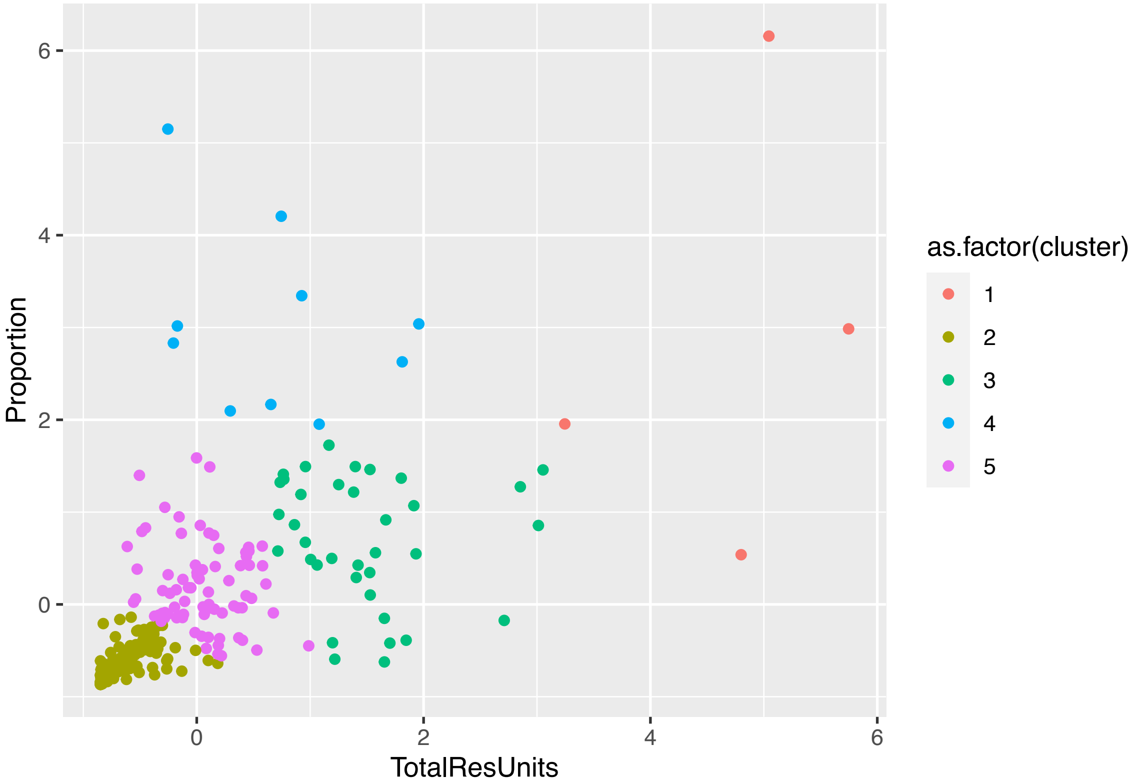
```r
#finding the optimal number of clusters using the elbow method
k.max <- 15
data <- zscores3
wss <- sapply(1:k.max,
              function(k){kmeans(data, k)$tot.withinss})

wss
```

```
##  [1] 990.99617 459.68133 228.76635 222.07259 156.59283 155.52934 154.41252
##  [8] 153.95372  81.51228  77.19872 138.37655  80.28117  80.42741  72.68148
## [15]  75.96500
```

```r
plot(1:k.max, wss,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")
```




```r
ggplot(zScores3)+
  geom_point(aes(x=TotalResUnits, y=Proportion, color=as.factor(cluster)))
```



#Q3:Choose one other neighborhood and compare the average residential property costs with your neighborhood.

```
#Comparing the revenue and cost of Madison and Long Island

NYC_Madison <- filter(NYCdf,NEIGHBORHOOD_ID=="149", Year> 2008) %>%
  group_by(Year) %>%
  select ("SALE_PRICE", "RESIDENTIAL_UNITS", "COMMERCIAL_UNITS")
```

```
## Adding missing grouping variables: `Year`
```

```
Revenue_Madison <- summarise(NYC_Madison, Revenue = sum(SALE_PRICE, na.rm = T), TotalResUnits=sum(RESIDI
  filter (Revenue > 0) %>%
  drop_na() %>%
  mutate (AveragePropertyCost=Revenue/TotalResUnits)

NYC_LongIsland <- filter(NYCdf,NEIGHBORHOOD_ID=="147", Year>2008) %>%
  group_by(Year)%>%
  select ("SALE_PRICE", "RESIDENTIAL_UNITS", "COMMERCIAL_UNITS")
```

```
## Adding missing grouping variables: `Year`
```

```
Revenue_LongIsland <- summarise(NYC_LongIsland, Revenue = sum(SALE_PRICE, na.rm = T), TotalResUnits=sum
  filter (Revenue > 0) %>%
  drop_na()%>%
```

```r
  mutate (AveragePropertyCost=Revenue/TotalResUnits)

#descriptivestatistics_Madison
summarise(Revenue_Madison, MadisonMeanRevenue = mean(Revenue))
```

```
## # A tibble: 1 x 1
##   MadisonMeanRevenue
##                <dbl>
## 1          110506266
```

```r
#descriptivestatistics_LongIsland
summarise(Revenue_LongIsland, LIMeanRevenue = mean(Revenue))
```

```
## # A tibble: 1 x 1
##   LIMeanRevenue
##           <dbl>
## 1    205165453.
```

```r
#t test
t.test(x= Revenue_Madison$AveragePropertyCost, y=Revenue_LongIsland$AveragePropertyCost, alternative =
```

```
##
##  Welch Two Sample t-test
##
## data:  Revenue_Madison$AveragePropertyCost and Revenue_LongIsland$AveragePropertyCost
## t = -1.4981, df = 10.506, p-value = 0.1635
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -747916.2  144181.0
## sample estimates:
## mean of x mean of y
##  562827.7  864695.4
```