Assignment8

Haoran Xu

2024-11-17

```
# libraries
library(here)
```

```
## here() starts at /Users/horanxu/Desktop/GEOG714_Applied_Data_Analysis_for_Geographers_and_Earth_Scie
```

```
library(readxl)
```

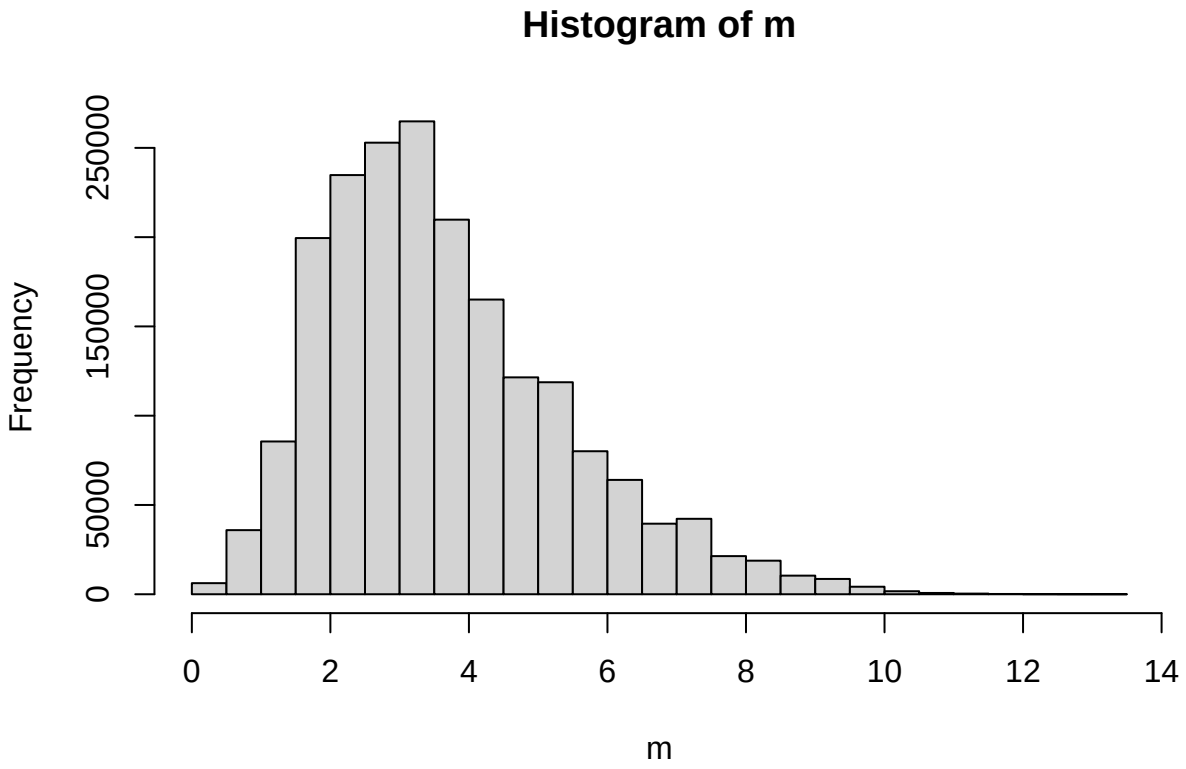
```
df <- data.frame(read_excel(paste0(here(), "/Assignments/Assignment8/Canada2006_WVS.xlsx")))
df_cleaned <- df[apply(df, 1, function(x) !any(x < 0)), ]
df2 <- as.data.frame(apply(df_cleaned[, 2:7], 2, function(x) as.numeric(x)))
names(df2) <- c("v1", "v2", "v3", "v4", "v5", "v6")

summary(df2)
```

```
##        v1              v2              v3              v4
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   : 1.000
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.: 7.000
##  Median :2.000   Median :2.000   Median :1.000   Median : 8.000
##  Mean   :1.584   Mean   :1.822   Mean   :1.372   Mean   : 7.762
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.: 9.000
##  Max.   :4.000   Max.   :4.000   Max.   :3.000   Max.   :10.000
##        v5              v6
##  Min.   :0.0000   Min.   : 1.00
##  1st Qu.:0.0000   1st Qu.: 6.00
##  Median :0.0000   Median : 7.00
##  Mean   :0.2151   Mean   : 7.13
##  3rd Qu.:0.0000   3rd Qu.: 8.00
##  Max.   :2.0000   Max.   :10.00
```

```
print(sapply(df2, sd))
```

```
##        v1        v2        v3        v4        v5        v6
## 0.5962391 0.8174581 0.5477178 1.6976487 0.5351201 1.9658523
```

```
m <- dist(df2, diag=TRUE)
hist(m)
```




```
summary(m)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.449   3.317   3.707   4.690  13.454
```

Q1. Write code that: a. Creates a vector of ten elements each with a value equal to 1 b. Loops through the elements one at a time subtracting the value of i from the value of the element in the ith position. Use a for loop for this task.

```
v <- c(rep(1, 10))
for (i in 1:length(v)){
  v[i] <- v[i] - i
}
```

```
v <- c(0, 0, 0, 0, 0)
vv <- rep(v, 5)
k <- 0
for (i in 1:length(v)){
  for (j in 1:length(v)){
    k <- k + 1
    vv[k] <- i * j
  }
```

```
}
vv
```

```
##  [1]  1  2  3  4  5  2  4  6  8 10  3  6  9 12 15  4  8 12 16 20  5 10 15 20 25
```

Q2. What does the vector vv contain?

The vector vv contains 25 elements, which are "1 2 3 4 5 2 4 6 8 10 3 6 9 12 15 4 8 12 16 20 5 10 15 20 25".

Q3. Assume mm is a 5 by 5 matrix (mm <- matrix(0,nrow=5,ncol=5). Write code to fill it with values from 1 to 25 using a nested for loop?

```
mm <- matrix(0, nrow = 5, ncol = 5)
k <- 0
for (i in 1:nrow(mm)){
  for (j in 1:ncol(mm)){
    k <- k + 1
    mm[i, j] <- k
  }
}
```

```
d <- matrix(0, nrow = nrow(df2), ncol = nrow(df2))
mm <- as.matrix(df2)

for(i in 1:nrow(df2)){
  for(j in 1:nrow(df2)){
    d[i,j] <-
      sqrt((mm[i,1]-mm[j,1])**2) +
      sqrt((mm[i,2]-mm[j,2])**2) +
      sqrt((mm[i,3]-mm[j,3])**2) +
      sqrt((mm[i,4]-mm[j,4])**2) +
      sqrt((mm[i,5]-mm[j,5])**2) +
      sqrt((mm[i,6]-mm[j,6])**2)
  }
}
```

Q4. Briefly describe the values of d[i,j]. What do they measure/represent?

The matrix of `d` is the distance matrix of `1994*1994` to measure the distances between the 1994 observations. The d[i, j] in the for loop represents calculating the distance between the i-th and j-th observations in `mm` matrix and assign the value to the i-th row and j-th column in `d` matrix.

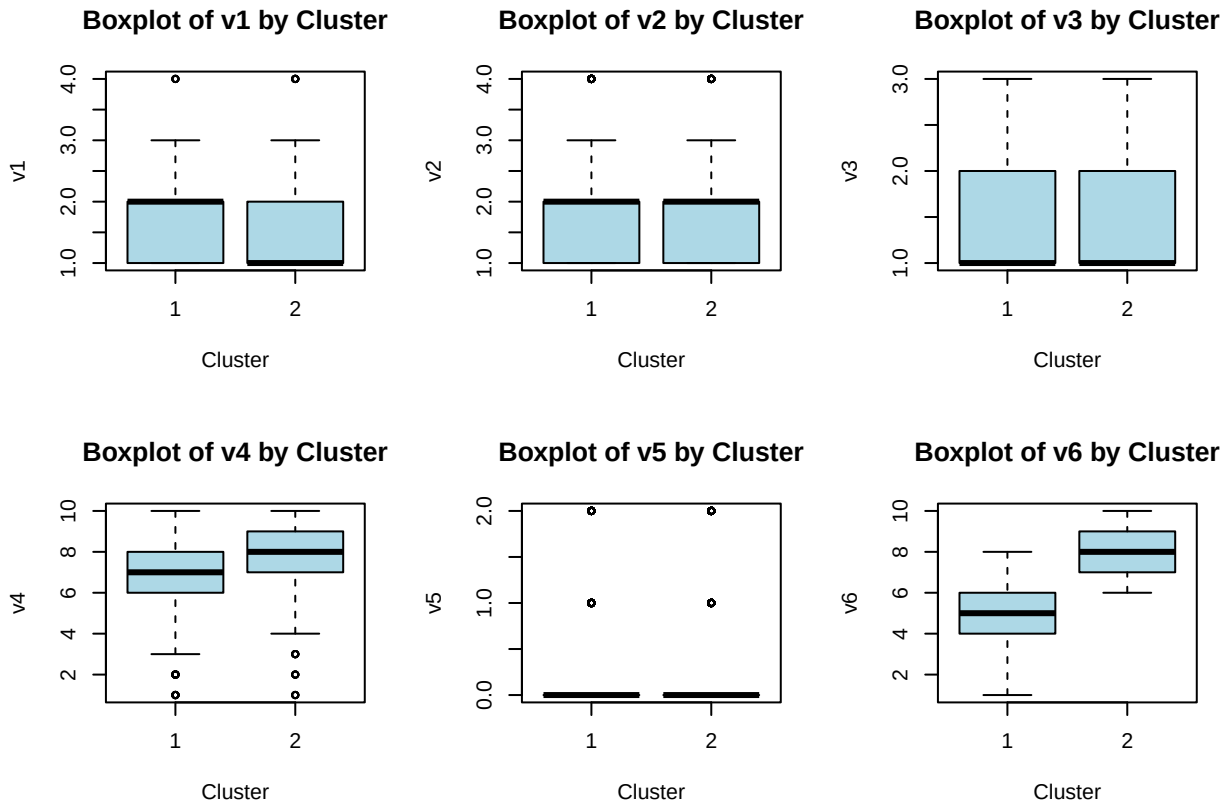
```
out1 <- kmeans(
  df2, centers = 2, iter.max = 50, nstart = 50, algorithm = "Lloyd"
)
out1$centers
```

```
##         v1       v2       v3       v4        v5       v6
## 1 1.715729 1.969697 1.419913 7.082251 0.2626263 5.023088
## 2 1.513451 1.743274 1.346656 8.123751 0.1898540 8.252882
```

```r
df2$member <- out1$cluster
```

Q5. Using the group assignment indicator (called 'member') you've merged to the data frame df, explore the data and do some analysis to understand what features members of the two groups (1 & 2) make them distinct from one another. Ensure that you use some visualisation (base R only) to assist you in your analysis. Complete this analysis in no more than four sentences and one or two graphics.

```r
group_means <- aggregate(. ~ member, data = df2, mean)

par(mfrow = c(2, 3))
for (v in names(df2)[1:6]) {
  boxplot(df2[[v]] ~ df2$member,
          main = paste("Boxplot of", v, "by Cluster"),
          xlab = "Cluster",
          ylab = v,
          col = "lightblue")
}
```



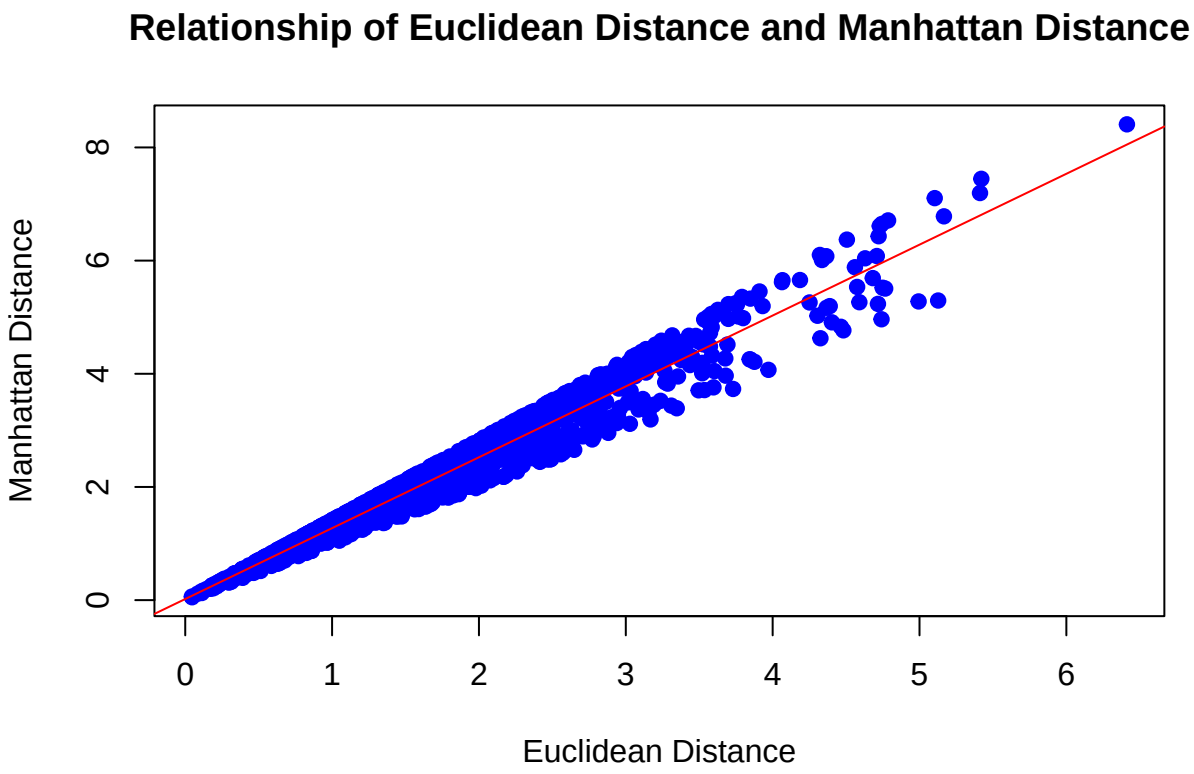
According to the mean calculated respectively for the two groups in `group_means`, the variations in means of v6 and v4 are the largest, with around 3.23 and 1.04 respectively. The box plots of the six varibales demonstrated the pattern very well, with v6 and v4 demonstrating the biggest variations. Tracing back to the descriptive analysis of `df2`, v6 and v4 both range from 1 to 10 (while the others mostly range from 1 to

4), and also have the highest standard deviations with 1,97 and 1.70, which contributed to the most of how Lloyd algorithm was used to grouping.

```
new_data <- function(obs) {
  the_data <- matrix(0, nrow = obs, ncol = 2)
  for (i in 1:obs){
    the_data[i, 1] <- rnorm(1)
    the_data[i, 2] <- rnorm(1)
  }
  return(the_data)
}

m <- data.frame(new_data(50))

d <- dist(m, method = "euclidean")
d2 <- dist(m, method = "manhattan")
```

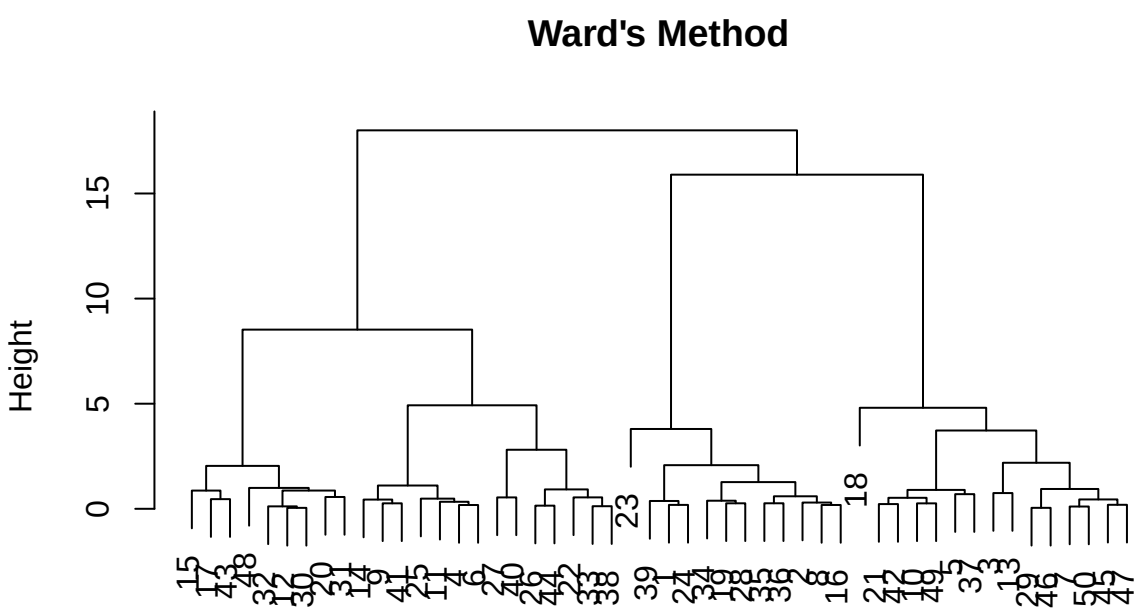
Q6. Create a scatterplot of these two distance matrices. Briefly comment on the pattern you see, and describe why you may be seeing this pattern (in less than 2 sentences). Hint: you may need to read up on these different distance metrics.

```
d_vec <- as.vector(d)
d2_vec <- as.vector(d2)

plot(d_vec, d2_vec,
     xlab = "Euclidean Distance",
     ylab = "Manhattan Distance",
     main = "Relationship of Euclidean Distance and Manhattan Distance",
     pch = 19, col = "blue")

abline(lm(d2_vec ~ d_vec), col = "red")
```

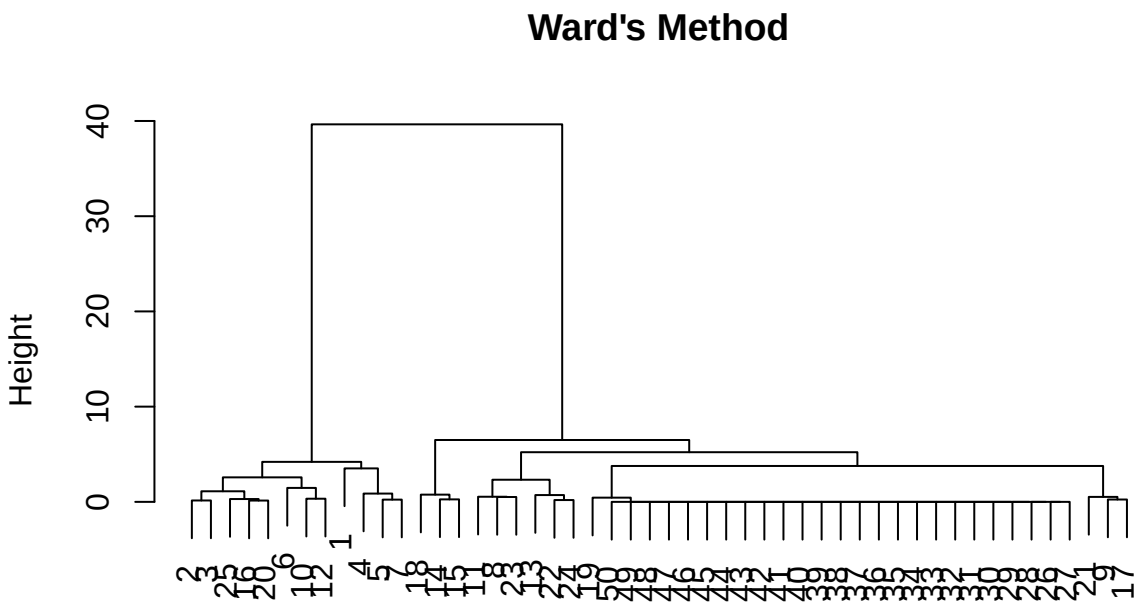


The plot shows a strong positive correlation between Euclidean and Manhattan distances, as most points align closely with the trend line and within a strict polygon area. However, Manhattan distance is mostly larger than Euclidean distance.

```
fit <- hclust(d, method = "ward.D")
plot(fit, main = "Ward's Method")
```




```
new_data2 <- function(obs) {
  the_data <- matrix(0, nrow = obs, ncol = 2)
  for (i in 1:obs/2){
    the_data[i, 1] <- rnorm(1)
    the_data[i, 2] <- rnorm(1)
  }
  for (i in obs/2:obs){
    the_data[i, 1] <- rnorm(1) - 2
    the_data[i, 2] <- rnorm(1) - 2
  }
  return(the_data)
}

m <- data.frame(new_data2(50))
d3 <- dist(m, method = "euclidean")
fit2 <- hclust(d3, method = "ward.D")
plot(fit2, main = "Ward's Method")
```

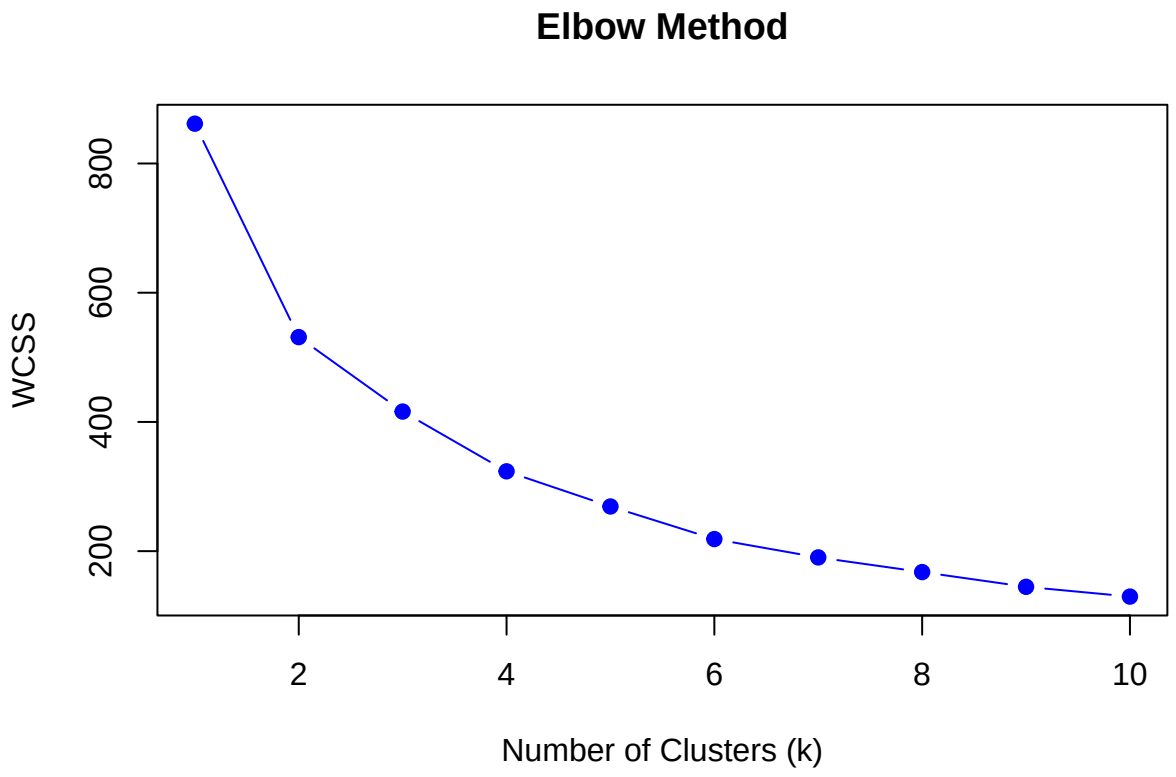


Q7. Use principal components analysis to reduce the data to fewer dimensions (PCs) that capture essential variation in the data, and then use k-means and/or hierarchical clustering to explore the composition of these rock samples using these PCs. Describe what you can learn from this process.

```
df3 <- data.frame(read_excel(paste0(here(), "/Assignments/Assignment8/Whole rock major oxide components

pca_out <- prcomp(df3[, 2:11],
                  center=TRUE, scale=TRUE)
summary(pca_out)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.2762 1.3679 0.99039 0.81875 0.68961 0.54276 0.46631
## Proportion of Variance 0.5181 0.1871 0.09809 0.06703 0.04756 0.02946 0.02174
## Cumulative Proportion  0.5181 0.7052 0.80329 0.87033 0.91788 0.94734 0.96909
##                           PC8     PC9      PC10
## Standard deviation     0.42223 0.36172 0.0001539
## Proportion of Variance 0.01783 0.01308 0.0000000
## Cumulative Proportion  0.98692 1.00000 1.0000000
```

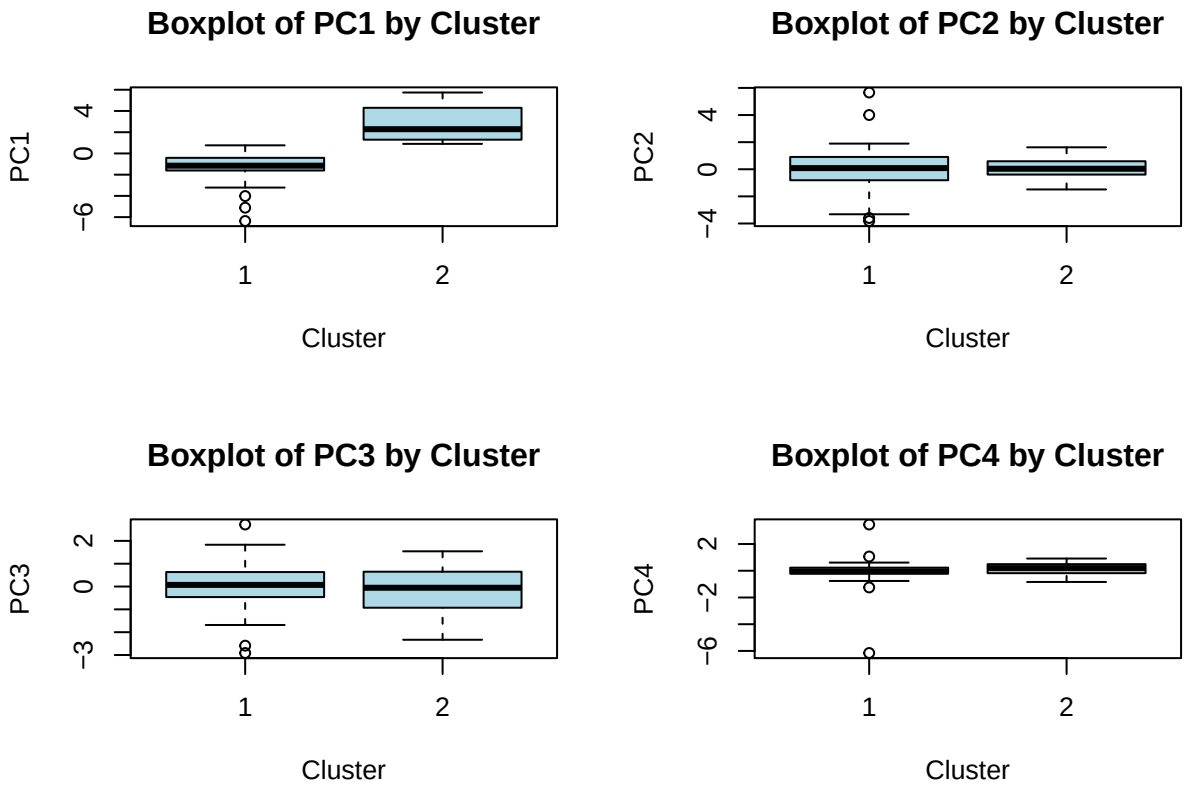
```r
key_pcs <- pca_out$x[,1:4]

# calculate total within-cluster sum of squares (WCSS)
wcss <- sapply(1:10, function(k) {
  kmeans(key_pcs, centers = k, nstart = 50)$tot.withinss
})

# draw Elbow Plot
plot(1:10, wcss, type = "b",
     xlab = "Number of Clusters (k)",
     ylab = "WCSS",
     main = "Elbow Method",
     pch = 19, col = "blue")
```




```r
kmeans_out <- kmeans(
  key_pcs, centers = 2, iter.max = 50, nstart = 50, algorithm = "Lloyd"
)

kmeans_out$centers
```

```
##         PC1        PC2         PC3         PC4
## 1 -1.214110 -0.0474239  0.06541666 -0.06008944
## 2  2.702374  0.1055564 -0.14560483  0.13374746
```

```
key_pcs <- as.data.frame(key_pcs)
key_pcs$member <- kmeans_out$cluster

par(mfrow = c(2, 2))
for (v in names(key_pcs)[1:4]) {
  boxplot(key_pcs[[v]] ~ key_pcs$member,
          main = paste("Boxplot of", v, "by Cluster"),
          xlab = "Cluster",
          ylab = v,
          col = "lightblue")
}
```




```
pca_out$rotation[,1:4]
```

```
##               PC1          PC2          PC3          PC4
## SiO2n   0.4168681 -0.162426917 -0.06214681  0.064252416
## TiO2n  -0.3424521 -0.343454359 -0.16482306 -0.061140780
## Al2O3n -0.1641930  0.299028331  0.75670324 -0.391738537
## CaOn   -0.3948925  0.153455777 -0.01185091  0.186097501
## P2O5n  -0.3499406 -0.269952895 -0.18385044 -0.006173904
## MgOn   -0.2861857  0.326472275 -0.05202399  0.538259834
## FeOn   -0.3620259 -0.001145467 -0.17099063 -0.530420753
## Na2On   0.1265629 -0.528967355  0.50109297  0.275179431
## K2On    0.3572386  0.028129281 -0.26459004 -0.382079088
## MnOn   -0.2167650 -0.531631676  0.09686878 -0.105952733
```

The code first used PCA to reduce the dimensionality of the data, which makes the number of variables reduce from 10 different chemical attributes to only 4 principal components (I set the threshold of cumulative proportion of variances explained to be 90%). Then using the 4 principal components, k-means clustering analysis is conducted. To decide how many centroids/groups I should set, I introduced Elbow plot and picked the "elbow" point (k = 2) to perform k-means clustering.

Then I replicated the moves previously in Q5 to illustrate the four box plots of four PCs to compare their differences. And the results showed that the variation across groups in PC1 is the largest. Tracing back the loading of PC1 on 10 chemicals, I found that SiO2n, CaOn, FeOn, K2On, P2O5n, TiO2n have the biggerst loadings of an abosolute valueover of over 0.3. Therefore these chemicals in the rock samples might be the underlying and crucial factors that differs different rock types.