# Homework 6

*Cody Frisby*

*4/19/2017*

## 6.1

### Part A

Using the single linkage procedure with the distance matrix in table 1.

Table 1: Distance Matrix

|  | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 |  |  |  |
| 2 | **1** | 0 |  |  |
| 3 | 11 | 2 | 0 |  |
| 4 | 5 | 3 | 4 0 |  |

First step we group the two that are closest. That would be 1 and 2.

|  | 12 | 3 | 4 |
|---|---|---|---|
| 12 | 0 |  |  |
| 3 | **2** | 0 |  |
| 4 | 3 | 4 | 0 |

We repeat the last step using the new matrix. Here we would group 12 with 3.

|  | 123 | 4 |
|---|---|---|
| 123 | 0 |  |
| 4 | 3 | 0 |

And then we would group them all at 3, resulting in a single cluster.

### Part B

Still using table 1, using the complete linkage procedure first merging groups 1 and 2.

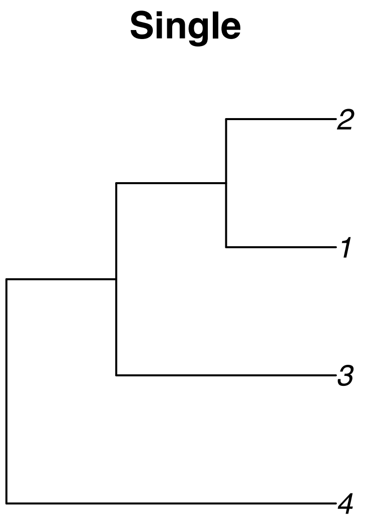
|  | 12 | 3 | 4 |
|---|---|---|---|
| 12 | 0 |  |  |
| 3 | 11 | 0 |  |
| 4 | 5 | **4** | 0 |

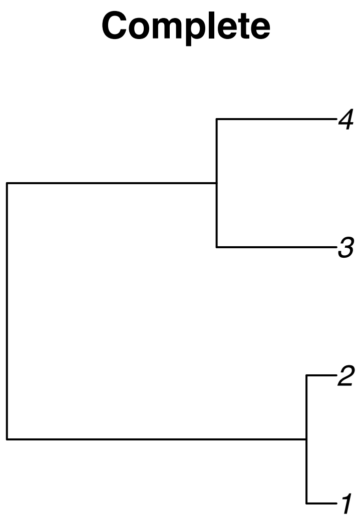
We now merge groups 3 and 4.

|  | 12 | 34 |
|---|---|---|
| 12 | 0 |  |
| 34 | 11 | 0 |

And finally merging all groups at 11.

**Part C**

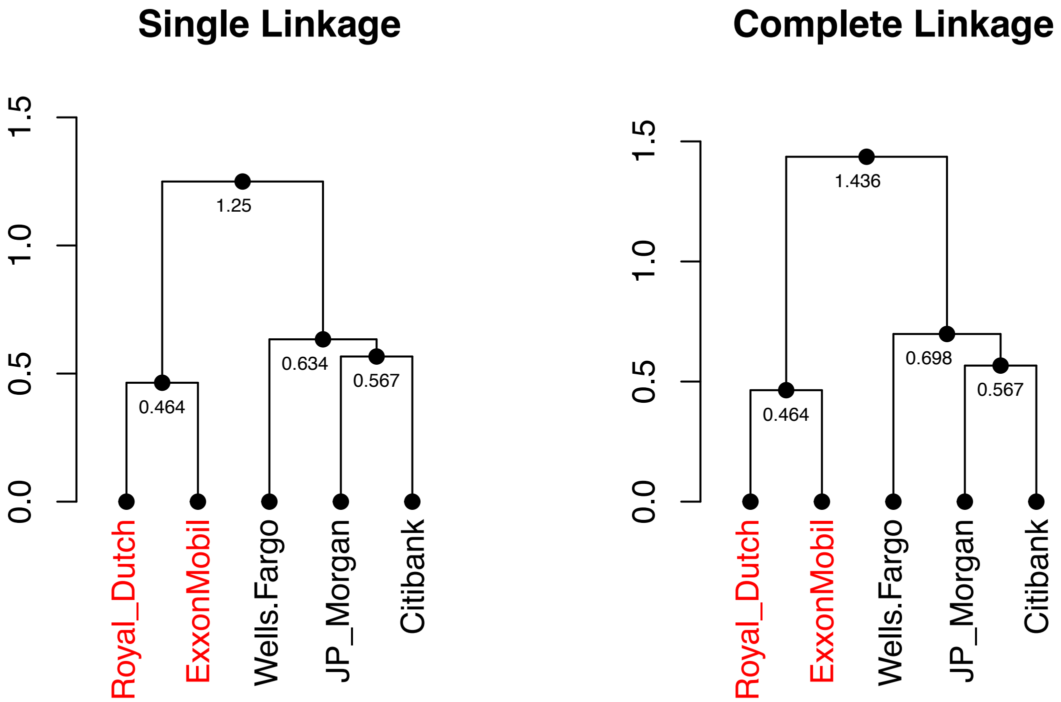




The complete has a more simple structure, merging 1 and 2 at 1, and 3 and 4 at 4. Single linkage procedure results in more total clusters.

### 6.2

I get the following results when clustering by **stocks**.



The two methods generate two very similar results as can be seen from the above plots. The only difference appears to be the height at which the clusters are joined at, which is to be expected based on the differences between single and complete linkage. We appear to have clustering according to member industries, financial and oil/gas.

For reference, I display the distance matrix in table 6.

Table 6: Distance matrix for stocks data

| | JP_Morgan | Citibank | Wells.Fargo | Royal_Dutch | ExxonMobil |
|---|---|---|---|---|---|
| JP_Morgan | 0.0000000 | 0.5665677 | 0.6980579 | 1.4322328 | 1.4360206 |
| Citibank | 0.5665677 | 0.0000000 | 0.6336421 | 1.2496081 | 1.3347625 |
| Wells.Fargo | 0.6980579 | 0.6336421 | 0.0000000 | 1.3584291 | 1.4022214 |
| Royal_Dutch | 1.4322328 | 1.2496081 | 1.3584291 | 0.0000000 | 0.4641351 |
| ExxonMobil | 1.4360206 | 1.3347625 | 1.4022214 | 0.4641351 | 0.0000000 |

### 6.3

Starting with the two clusters AB and CD from the data in the below table. To find the distance I use the formula

$$d = \sum_{i=1}^{2} (x_i - center)^2$$

where center is found as the mean of the $x_1$ points and $x_2$ points for any given cluster.

| | x1 | x2 |
|---|---|---|
| A | 5 | 4 |
| B | 1 | -2 |
| C | -1 | 1 |

|   | x1 | x2 |
|---|---|---|
| D | 3 | 1 |

The center of AB is $(3, 1)$ and the center of CD is $(1, 1)$. The distance of A to AB is $(5-3)^2 + (4-1)^2 = 13$ and A's distance to CD is $(5-1)^2 + (4-1)^2 = 25$. These are the results if A is NOT reassigned. If A is reassigned, its distance to B is 52 and to ACD is 11.11. A is reassigned to ACD.

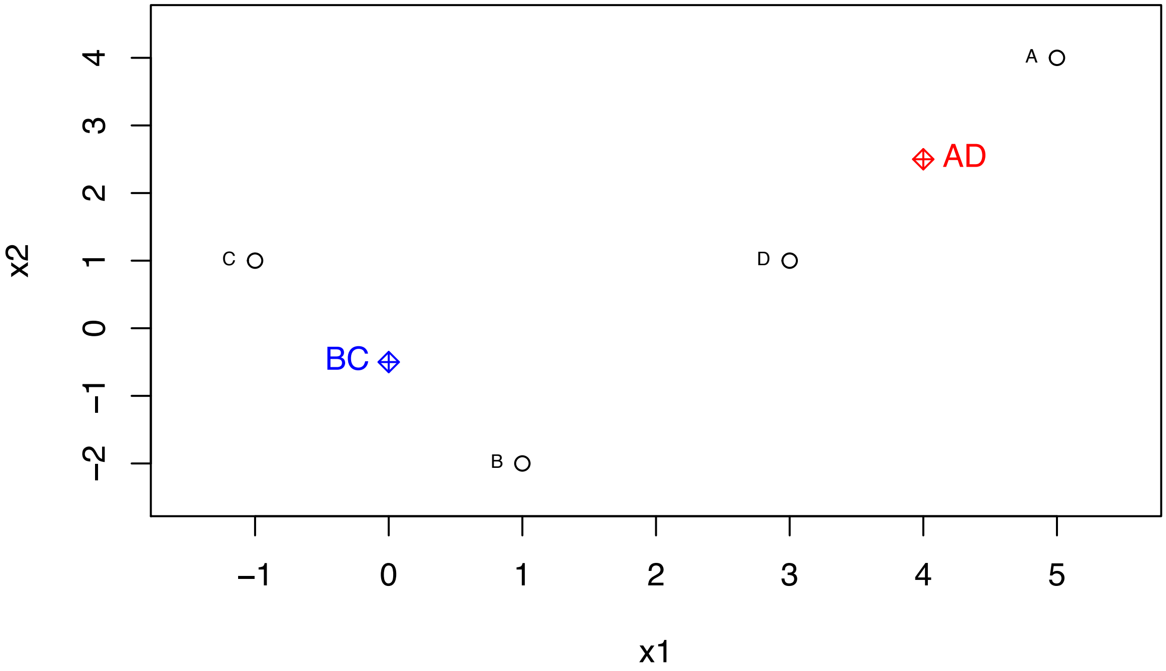
Checking B we do the same as with A except now B is by itself so its distance from itself is 0. If we reassigned B its distance to ABCD is 10. B is NOT reassigned.

If C is not moved, its distance to B is 13 and to ACD is 12.111. If C is moved its distance to BC is 3.25. C is reassigned to new cluster BC.

If D is not moved, its distance to AD is 3.25. If D is moved its distance to BCD is 5 while its distance to A is 13. D remains in AD.

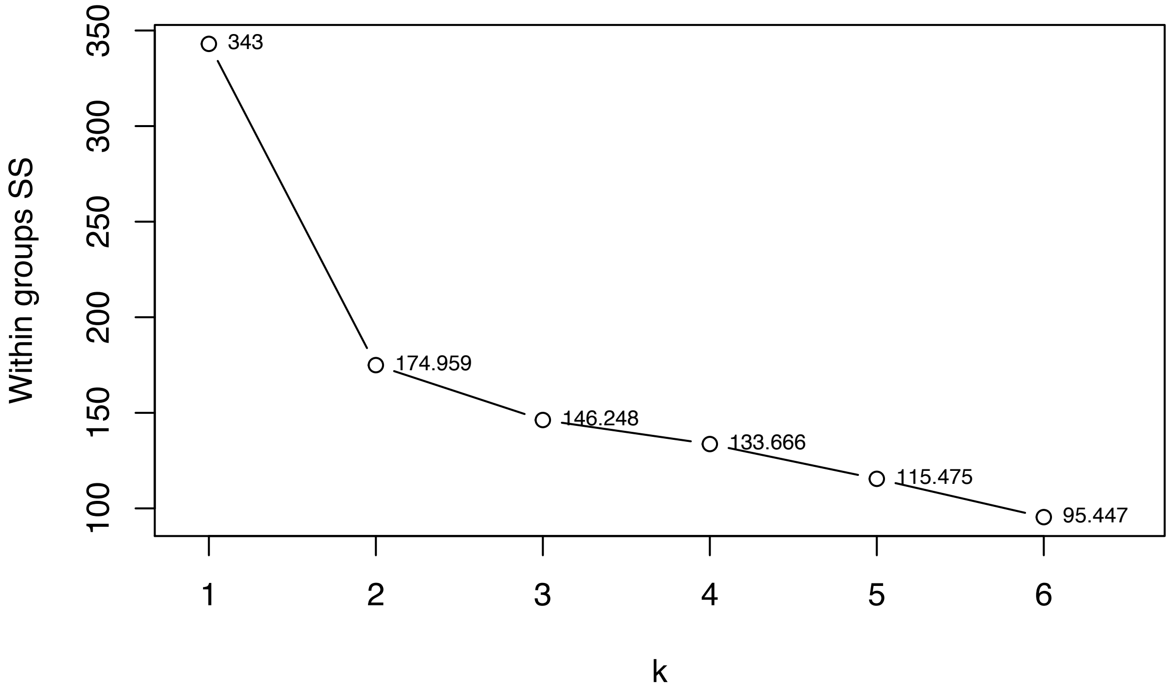
We conclude with clusters BC, centered at $(0, -\frac{1}{2})$, and AD, centered at $(4, 2.5)$.

I display a plot for reference.



### 6.4

Like the example in the book, I also will remove DC since it is an outlier for *murder* variable as well as being high in many of the other crime stats. Reproducing the same plot in the text except here I use the standard deviation on the data instead of the range.

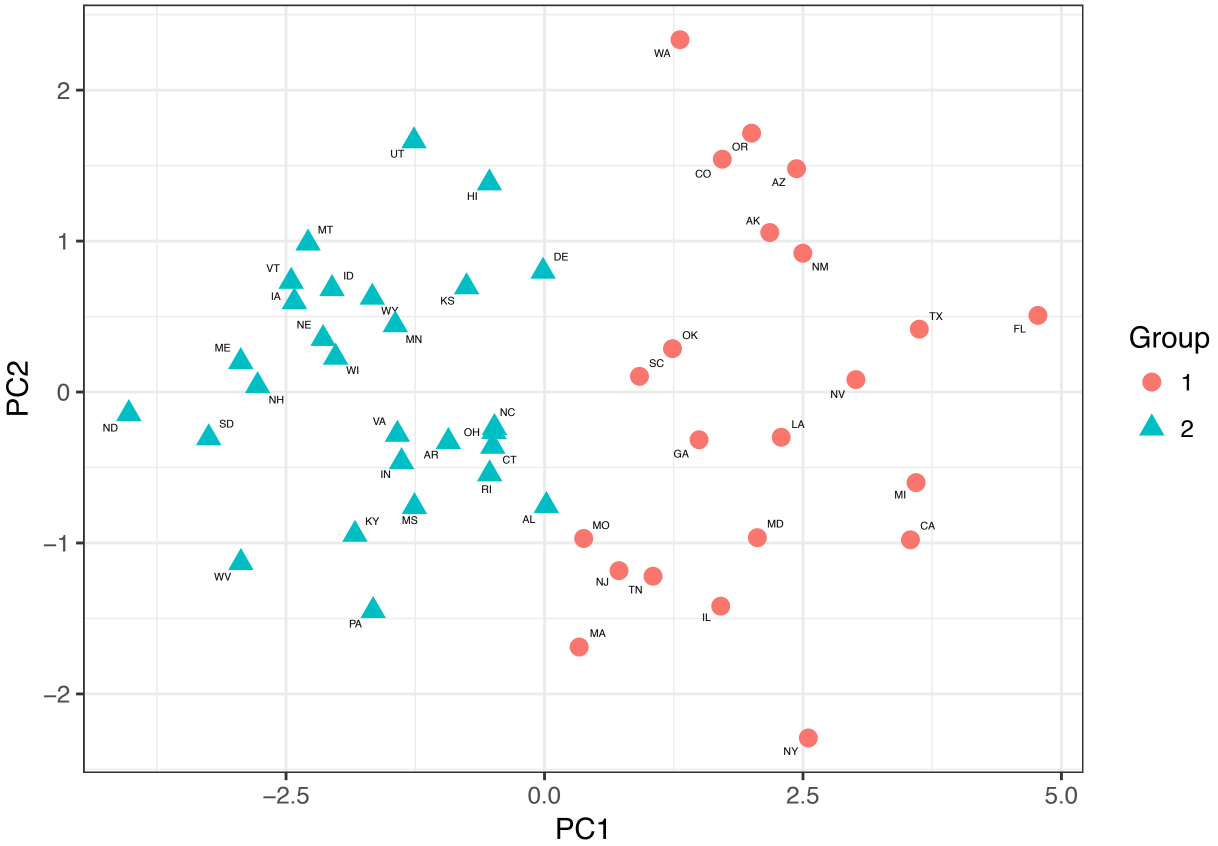


We can see that it somewhat flat lines after k = 2. Just like in the text, we look at a two-group solution.

Table 8: Means for two-group solution

| | Murder | Rape | Robbery | Assault | Burglary | Theft | Vehicle |
|---|---|---|---|---|---|---|---|
| 1 | 9.368182 | 341.5962 | 880.8041 | 594.91649 | 53.29901 | 623.5222 | 1986.7384 |
| 2 | 19.868629 | 238.3712 | 509.0143 | 4.51905 | 240.28620 | 1430.6561 | 247.0357 |

Below is a plot of the two-group solution using the first two principal components to visualize the two clusters. We can see the two distinct clusters with no points overlapping into the other cluster!



The results are very similar to those presenting in the book where they standardized each variable with the range. Table 8 displays each variables standard deviations for reference.

Table 9: Standard Deviations

| Murder | Rape | Robbery | Assault | Burglary | Theft | Vehicle |
|---|---|---|---|---|---|---|
| 3.454696 | 14.48321 | 109.0393 | 139.1888 | 419.3984 | 751.8488 | 209.7555 |

---

## R code:

```
#####  6.2
df <- read.table("~/Documents/STAT4400/data/stock-price.txt",
                 header = TRUE)
rownames(df) <- paste0("week", rownames(df))
### one way to work the problem:
#df <- t(df)
#d <- dist(df)
# another way to work the problem:
d <- dist(cor(df))
clust_s <- hclust(d, method = "single")
clust_c <- hclust(d)
# vector of colors
labelColors <- c(1:k)
```

```r
# cut dendrogram in 4 clusters
clusMember <- cutree(clust_s, k)
t_single <- table(clusMember)
# function to get color labels
colLab <- function(n) {
  if (is.leaf(n)) {
    a <- attributes(n)
    labCol <- labelColors[clusMember[which(names(clusMember) == a$label)]]
    attr(n, "nodePar") <- c(a$nodePar, lab.col = labCol)
  }
  n
}
# using dendrapply
clusDendro <- dendrapply(hcd, colLab)
# make plot
hcd <- as.dendrogram(clust_s)
par(mfrow = c(1, 2))
plot(clusDendro, main = "Single Linkage", type = "triangle")
# cut dendrogram in 4 clusters
clusMember <- cutree(clust_c, k)
t_complete <- table(clusMember)
hcd <- as.dendrogram(clust_c)
clusDendro <- dendrapply(hcd, colLab)
# make plot
plot(clusDendro, main = "Complete Linkage",
     type = "triangle")
#####  6.4
df <- read.csv("~/Documents/STAT4400/data/crime.csv")
row.names(df) <- df$X
df <- df[-1]
# remove the outlier, DC
df <- df[row.names(df) != "DC", ]
# standardize all variables:
df_sd <- apply(df, 2, function(x) x / sd(x))
v <- apply(df, 2, sd) # vector of scalars
wss <- rep(0, 6)
for (i in 1:6){
  wss[i] <- sum(kmeans(df_sd, centers = i)$withinss)
}
plot(wss, type = "b", xlab = "k", ylab = "Within groups SS",
     xlim = c(0.9, 6.5))
text(1:7, wss, pos = 4, round(wss,3), cex = 0.65)
km <- kmeans(df_sd, centers = 2)
means <- km$centers * v
knitr::kable(means)
# create the two-dimensional space with prcomp
pc <- prcomp(df_sd)
# let's try a different plot than the one in the book
dt <- data.frame(pc$x[,1], pc$x[,2], rownames(df_sd))
dt$Group <- ifelse(km$cluster == 1, "1", "2")
names(dt) <- c("x", "y", "z", "Group")
library(ggplot2); library(ggrepel)
gg <- ggplot(data = dt, aes(x = x, y = y)) + theme_bw()
gg <- gg + geom_text_repel(aes(label = z), size = 1.5,
                             segment.alpha = 0.5,
                             segment.size = 0.25)
gg <- gg + geom_point(aes(shape = Group, colour = Group), size = 3)
gg <- gg + xlab("PC1") + ylab("PC2")
gg
# table of standard deviations
temp <- t(as.matrix(v))
knitr::kable(temp)
```