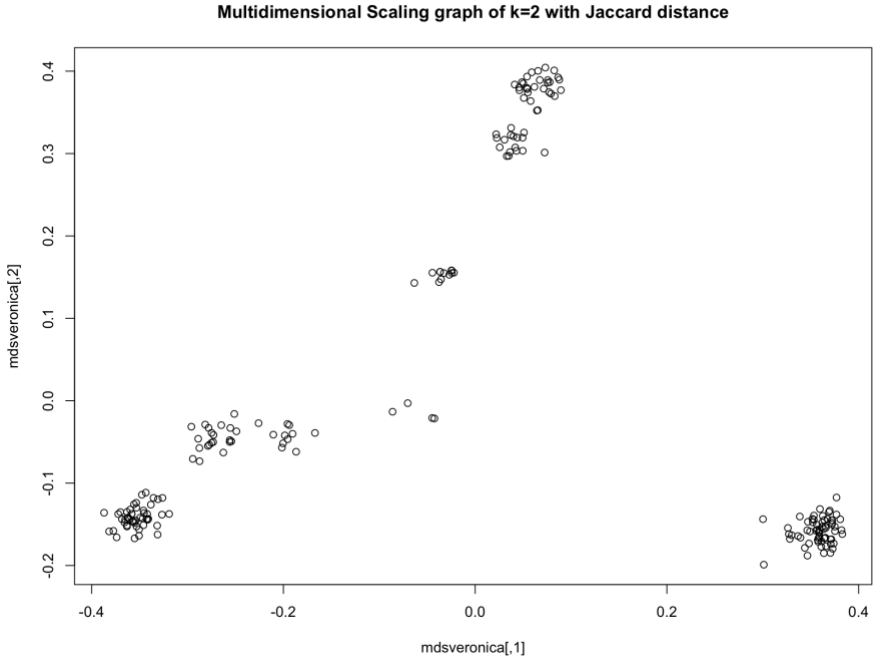
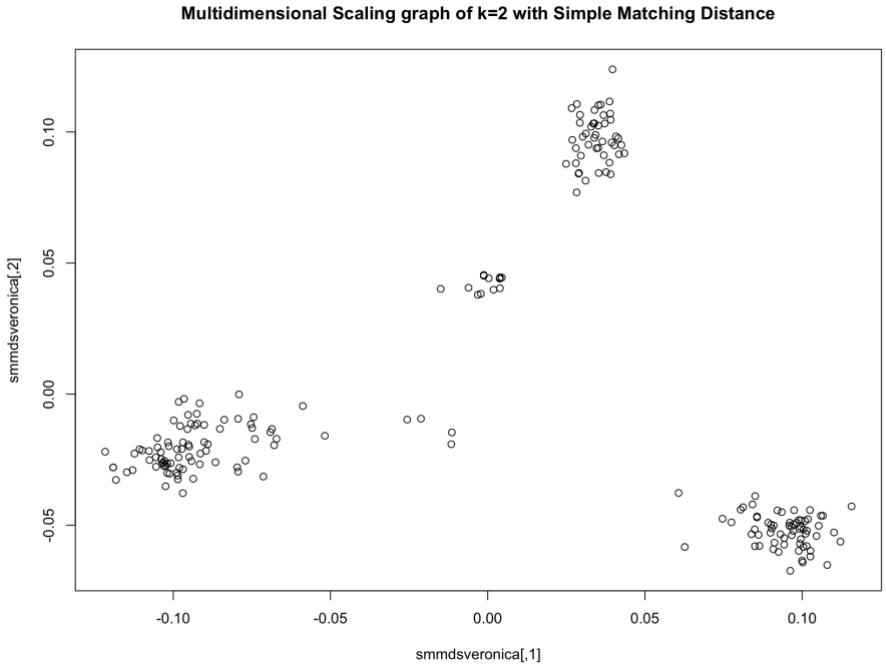
STAT 3019 Exercise 4: Yina Lin

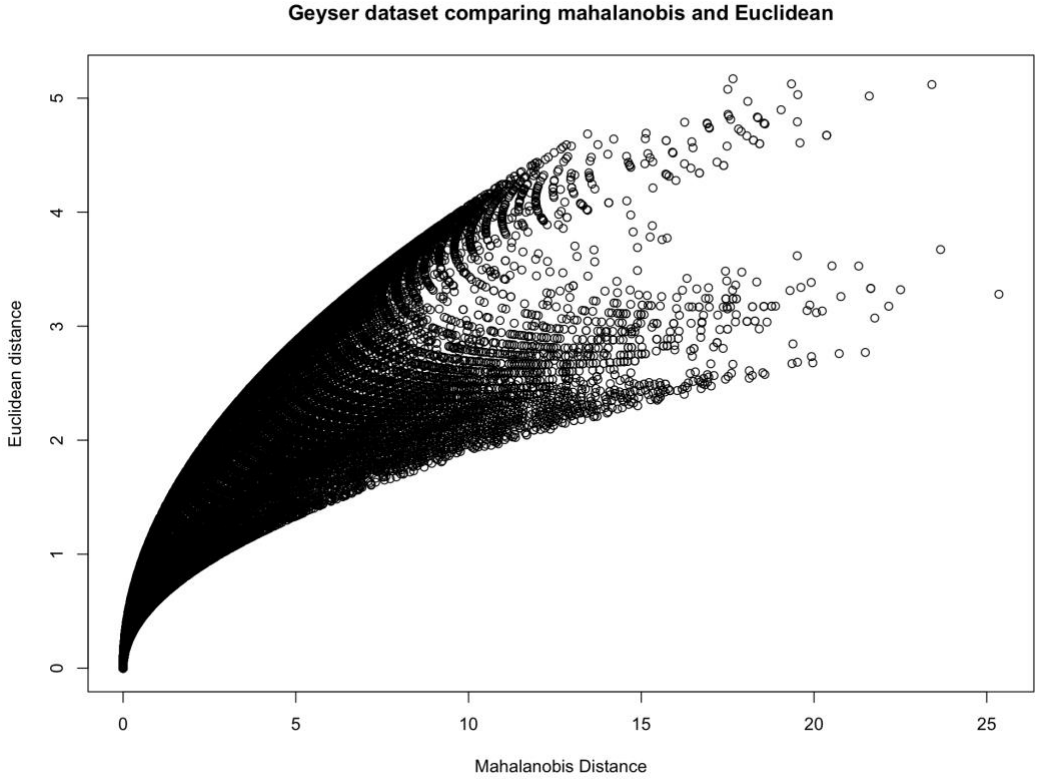
Q1: a. Produce a Multidimensional Scaling graph for the Veronica data with the simple matching distance and compared it with the graph from using the Jaccard distance.

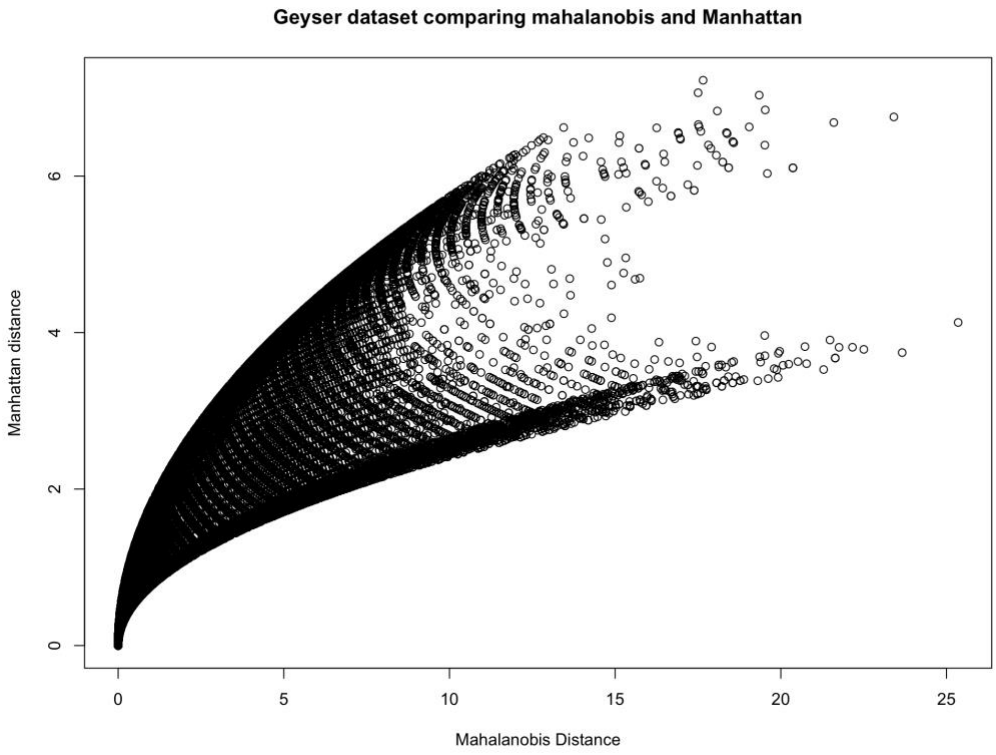


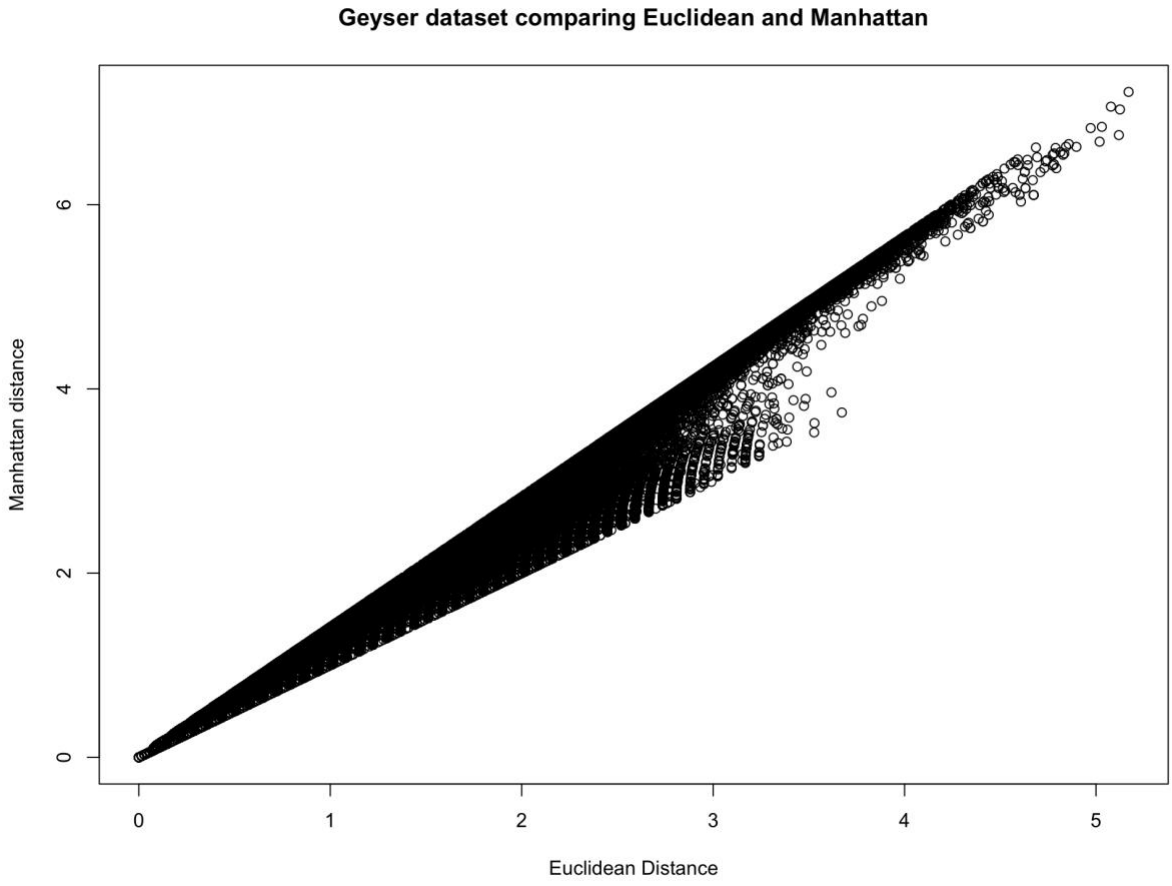


Here, since I do not know how to plot 2d+ plots with more coordinates. I used k=2 still, though you mentioned that k=2 might be weak. The graphs show that the dataset appears to be clearly clustered, with simple matching distance bit sparser than Jaccard distance.

b. for the Old Faithful Geyser dataset using Euclidean, Manhattan and Mahalanobis distance and compare them.







The Mahalanobis distance is quite different from both Euclidean and Manhattan distance. The shapes of Mahalanobis distance against Euclidean and Manhattan are quite similar. However, as the Mahalanobis distance increases, the Manhattan distance becomes averagely larger and has a wider range than Euclidean distances.

The Euclidean Distance vs. Manhattan distance plots shows that these two distances are fairly similar.

Codes used for Q1

```
## a. Veronica data
### Jaccard and simple matching distance
jveronica <- dist(veronica,method="binary")
smveronica <- dist(veronica,method="manhattan")/p  ## p=583
mdsveronica <- cmdscale(jveronica,k=2)   # jeronica
plot(mdsveronica, main="Multidimensional Scaling graph of k=2 with Jaccard distance")
smmdsveronica <- cmdscale(smveronica, k=2) # Simple matching
plot(smmdsveronica, main = "Multidimensional Scaling graph of k=2 with Simple Matching
Distance")
## b. Geyser data
eugeyser<- dist(sgeyser, method = "euclidean")
mgeyser <- dist(sgeyser, method = "manhattan")
str(geyser)   # 299 obs. of 2 variables
mahalm <- matrix(0,ncol=299,nrow=299)
geysercov <- cov(geyser)
mgeyser <- as.matrix(geyser)
for (i in 1:299)
  mahalm[i,] <- mahalanobis(mgeyser,mgeyser[i,],geysercov)
plot(as.dist(mahalm), eugeyser, main = "Geyser dataset comparing mahalanobis and Euclidean",
    xlab="Mahalanobis Distance", ylab = "Euclidean distance")
plot(as.dist(mahalm), mgeyser, main = "Geyser dataset comparing mahalanobis and Manhattan",
    xlab = "Mahalanobis Distance", ylab = "Manhattan distance")
plot(eugeyser, mgeyser, main = "Geyser dataset comparing Euclidean and Manhattan",
    xlab = "Euclidean Distance", ylab = "Manhattan distance")
```

Q2: a. Choosing K=9 for the unscaled Olive oil data compute eight K-means clusterings leaving out each single one of the variables. Compute the adjusted Rand index comparing each of these clusterings with the clustering computed on all variables.

There are 8 variables for the olive dataset. Therefore, here I wrote a loop for calculating the ARI for each pair.

```
library(mclust)
solive <- scale(olive)
olive9<- kmeans(olive, centers = 9, nstart = 100)
olive.wt1 <- kmeans(olive[,-1],centers=9,nstart=100)
adjustedRandIndex(olive9$cluster,olive.wt1$cluster)
ARI.olive<- array(NA,8)
i = 0

while (i < 8){
  i= i+1
  olive.wt <- kmeans(olive[,-i], centers = 9, nstart = 100)
  ARI.olive[i] <- adjustedRandIndex(olive9$cluster, olive.wt$cluster)
}
```

```
> ARI.olive
[1] 0.8498118 0.9923268 0.9968620 0.6013779 0.5850681 0.9965672 0.9776784 1.0000000
```

b. Do the same for the scaled Olive Oil data.

Same loop for the scaled olive data.

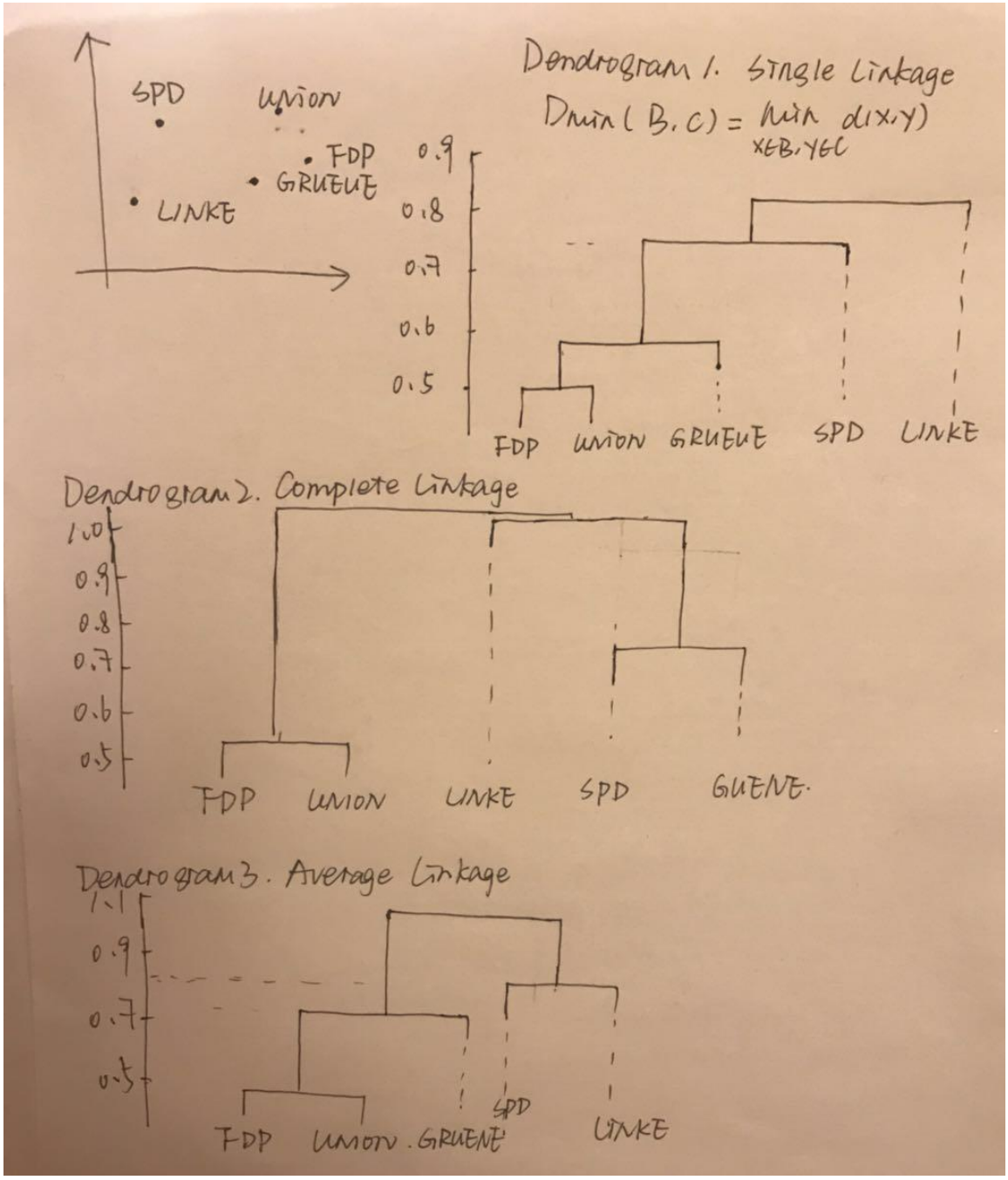
```
olive9s <- kmeans(solive,9,nstart=100)
ARI.solive<- array(NA,8)
i = 0
while (i < 8){
  i= i+1
  olive.wt <- kmeans(solive[,-i], centers = 9, iter.max=20, nstart = 100)
  ARI.solive[i] <- adjustedRandIndex(olive9s$cluster, olive.wt$cluster)
}
```

```
> ARI.solive
[1] 0.8929597 0.8904372 0.7105693 0.8996903 0.7650089 0.9573455 0.8720061 0.8940931
```

The results show that, for the unscaled olive dataset, removing the last variable, which is eicosenoic, has clustering completely identical to the clustering computed on all variables. Similar good results were shown for removing the 2nd, 3rd, 6th, and 7th variables as well.
 # 1 is somewhat strange result to me. But reasonable, the clusterings are strongly influenced by those variables with larger variations. It might be either due to the reason that the measurements of these variables are different, or that these variables with large ARI results are too small with fairly low variation, and indeed do not need to be considered for clustering. However, due to the fact that there are 5 variables indicating similar results, we might not just remove them.

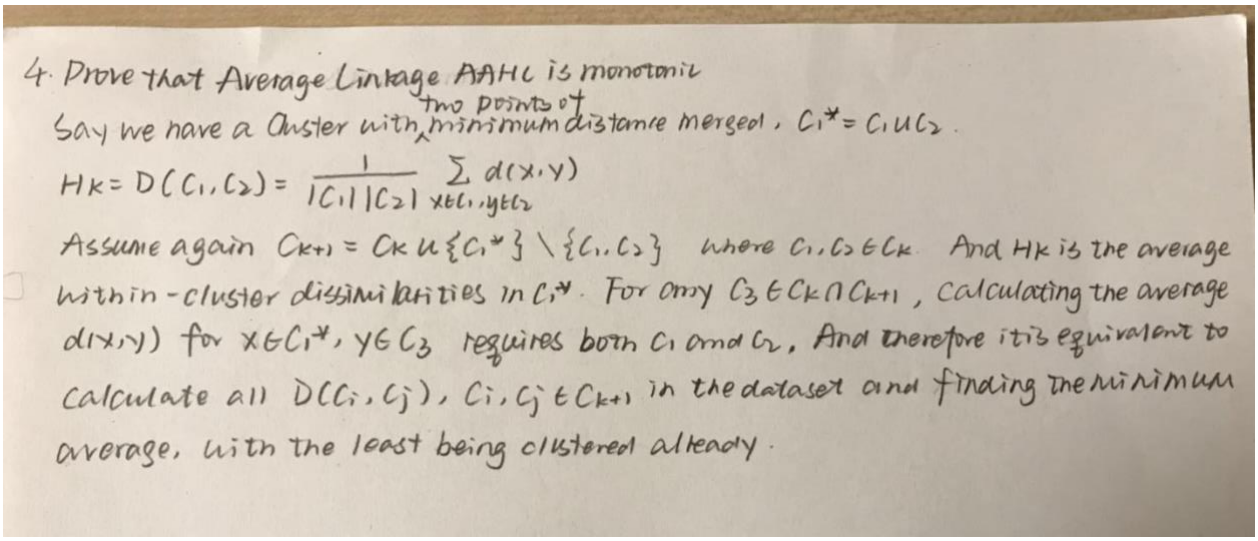
For the scaled olive data, we can see that all the variables show good results, among which removing the 6th variable, which is linolenic performs the best as identical datasets.

Q3: For the five parties in the Bundestag dataset, construct the dendrograms with Single Linkage, Complete Linkage and Average Linkage manually (without using the computer) and compare them.



Single Linkage produced minimal spanning trees and clusters points that are closed somewhere, whereas complete linkage avoids elongated clusters and gives low dissimilarities only two clusters that are close everywhere. Average linkage seems like a neutralized version of the two linkages.

Q4: Prove that Average Linkage AAHC is monotonic



4. Prove that Average Linkage AAHC is monotonic

Say we have a cluster with minimum distance (two points of) merged, $C_i^* = C_1 \cup C_2$.

$$H_k = D(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{x \in C_1, y \in C_2} d(x,y)$$

Assume again $C_{k+1} = C_k \cup \{C_i^*\} \setminus \{C_1, C_2\}$ where $C_1, C_2 \in C_k$. And H_k is the average within-cluster dissimilarities in C_i^*. For any $C_3 \in C_k \cap C_{k+1}$, calculating the average $d(x,y)$ for $x \in C_i^*$, $y \in C_3$ requires both C_1 and C_2, And therefore it is equivalent to calculate all $D(C_i, C_j)$, $C_i, C_j \in C_{k+1}$ in the dataset and finding the minimum average, with the least being clustered already.

Q5: For the Veronica dataset, compute Single Linkage, Complete Linkage and Average Linkage clusterings for a range of values of K including K = 8 (e.g., K = 2,...,20). For each K, measure the similarity between the three clusterings by averaging the three ARI-values that you get from comparing all pairs of clusterings.

```
### loop
single.veronica<- hclust(dist(veronica), method = "single")
complete.veronica<- hclust(dist(veronica), method = "complete")
average.veronica<- hclust(dist(veronica), method = "average")
aveARI<-c()
K=0
while (K < 20){
  K=K+1
single.veronica.k<- cutree(single.veronica, k=K)
complete.veronica.k<- cutree(complete.veronica, k=K)
average.veronica.k<- cutree(average.veronica, k=K)
x1<- adjustedRandIndex(single.veronica.k, complete.veronica.k)
x2<- adjustedRandIndex(single.veronica.k, average.veronica.k)
x3<- adjustedRandIndex(complete.veronica.k, average.veronica.k)
aveARI[K] <- sum(x1,x2,x3)/3
}
print(aveARI)
```

```
> print(aveARI)
 [1] 1.0000000 1.0000000 0.9520160 0.7890406 0.7518997 0.8716769 0.8617338 1.0000000 0.9893352 0.9841160 0.9833757
[12] 0.9377963 0.9131541 0.9121830 0.9114501 0.9159692 0.8021170 0.8033321 0.7995417 0.8008932
```

For k = 1,2,8. The average ARI-values are 1, the maximum, which includes 8, and seems to be a good method. However, the fact that K= 1 and 2 also give maximum value reveals something that the different methods used for AAHC might not affect much when K is small. It does not necessarily mean that this is a good number to cluster, where K=1 obviously is a

bad idea. In my opinion, it shows that when K=1,2 or 8, all the three linkages gives identical results of clustering based on their distances. So, for using the distance/dissimilarities method, say Jaccard distance, it is a good indicator of using K=8 has a stable result. However, it still did not allow us to test more results on different measurement on the distances.