Stat 218 - Cluster Analysis

Rommel Bartolome

March 22, 2019

1) Form clusters of the observations based on the four variables in the dataset. You may use either hierarchical or non-hierarchical (k-means) clustering. Show values and/or plots that support your final choice of clustering procedure. For k-means clustering, set a seed number.

We first load the necessary packages and the data itself.

```
library(tidyverse)
library(ISLR)
library(cluster)
```

Checking the `USArrests` dataset:

```
head(USArrests)
```

```
##            Murder Assault UrbanPop Rape
## Alabama      13.2     236       58 21.2
## Alaska       10.0     263       48 44.5
## Arizona       8.1     294       80 31.0
## Arkansas      8.8     190       50 19.5
## California    9.0     276       91 40.6
## Colorado      7.9     204       78 38.7
```

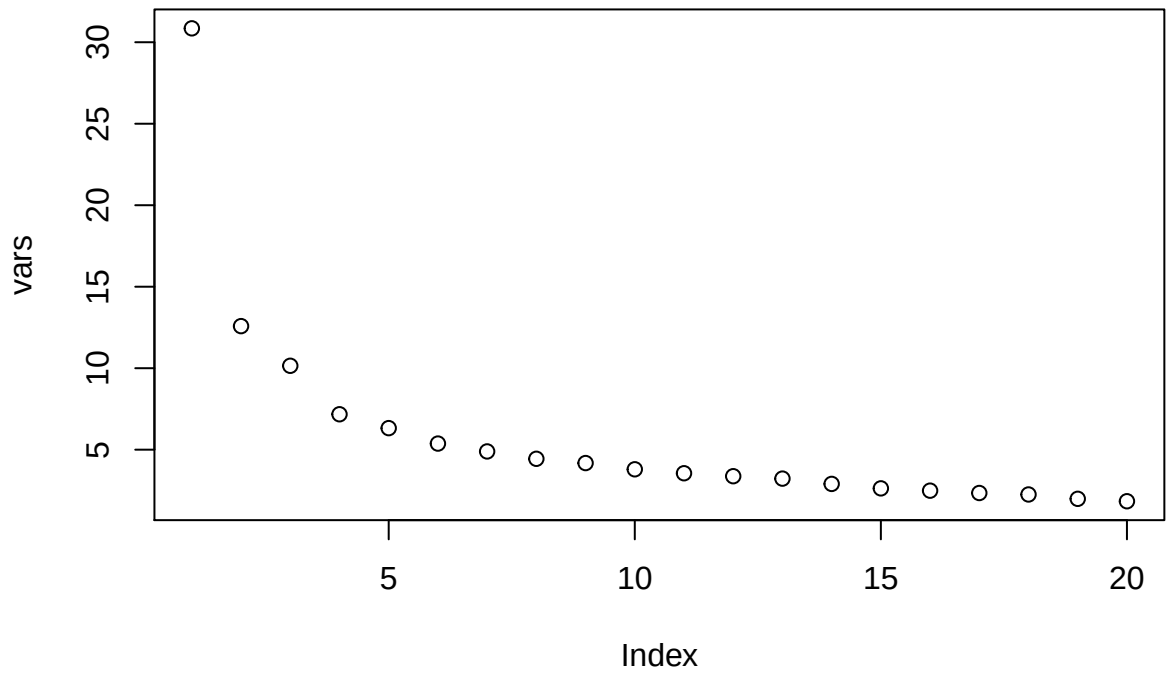
Here we see rows of cities, with four variables regarding USArrests. We will scale the data first based on standard deviation, then use k-means clustering initially. We will set 8 as the initial number of clusters, just because they say its lucky. The seed will be 15, my birthday.

```
seed <- 15
set.seed(seed)
sd.data <- scale(USArrests, F, T)
km.out <- kmeans(sd.data, 8)
table(km.out$cluster)
```

```
##
## 1 2 3 4 5 6 7 8
## 4 7 3 4 8 9 8 7
```

It appears that 8 clusters are okay as it is quite well distributed. However, we would like to determine a more appropriate cluster number by plotting the variation within the cluster versus k, and not basing it on a "lucky number".

```
vars <- rep(NA, 20)
for (i in 1:20){
  set.seed(seed)
  vars[i] <- kmeans(sd.data, i)$tot.withinss
}
plot(vars)
```



From the variability plot, it appears that 9 is quite a good number. We check the ditribution of the 9 cluster:

```
set.seed(seed)
km.out9 <- kmeans(sd.data, 9)
table(km.out9$cluster)
```

```
##
## 1 2 3 4 5 6 7 8 9
## 4 7 3 4 8 7 6 6 5
```

This appears 9 is *okay* too.

We check the variation, from 5 to 15 since it appears that it is where a reasonable "elbow" can be found:

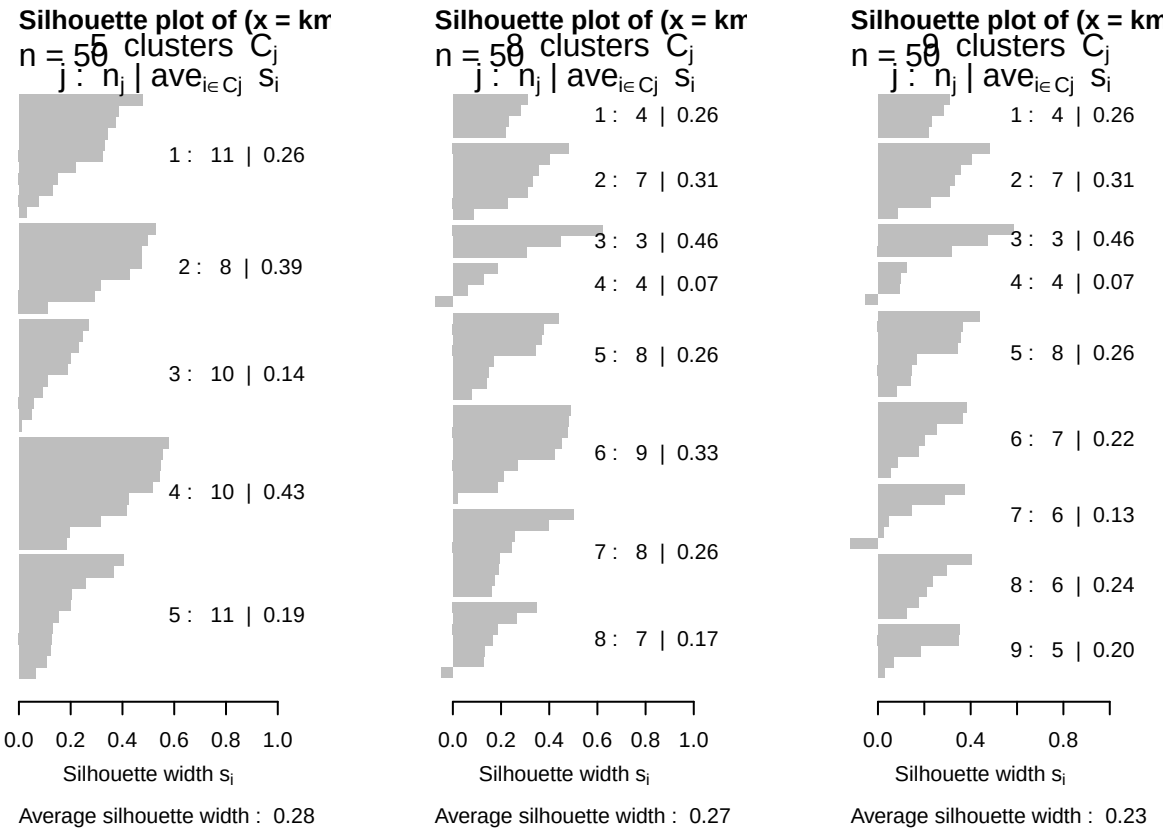
```
for (i in 3:15){
  set.seed(seed)
  km.out_n <- kmeans(sd.data, i)
  table(km.out_n$cluster) %>% print()
}
```

```
##
##  1  2  3
## 11  9 30
##
##  1  2  3  4
## 11  8 19 12
##
##  1  2  3  4  5
## 11  8 10 10 11
```

```
##
##  1  2  3  4  5  6
##  4  7  7 14  8 10
##
##  1  2  3  4  5  6  7
##  4  7  4  4  8 11 12
##
## 1 2 3 4 5 6 7 8
## 4 7 3 4 8 9 8 7
##
## 1 2 3 4 5 6 7 8 9
## 4 7 3 4 8 7 6 6 5
##
##  1  2  3  4  5  6  7  8  9 10
##  4  6  3  4  7  7 10  1  5  3
##
##  1  2  3  4  5  6  7  8  9 10 11
##  7  6  3  4  3  7  7  1  5  3  4
##
##  1  2  3  4  5  6  7  8  9 10 11 12
##  5  6  3  3  4  5  4  1  6  5  4  4
##
##  1  2  3  4  5  6  7  8  9 10 11 12 13
##  8  6  3  3  4  5  4  1  6  2  3  4  1
##
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14
##  5  5  3  4  3  2  7  1  4  5  4  2  1  4
##
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
##  4  5  3  4  3  2  7  1  4  2  3  2  3  4  3
```

From here, we see that 5, 8 and 9 clusters are good candidates. We check the silhoutte plot of 5, 8 and 9:

```r
par(mfrow = c(1,3))
for (n in c(5,8,9)){
  set.seed(seed)
  km.out <- kmeans(sd.data, n)
  data.dist <- dist(sd.data)
  plot(silhouette(km.out$cluster, data.dist))
}
```




```
km.out_fin <- kmeans(sd.data, 5)
```

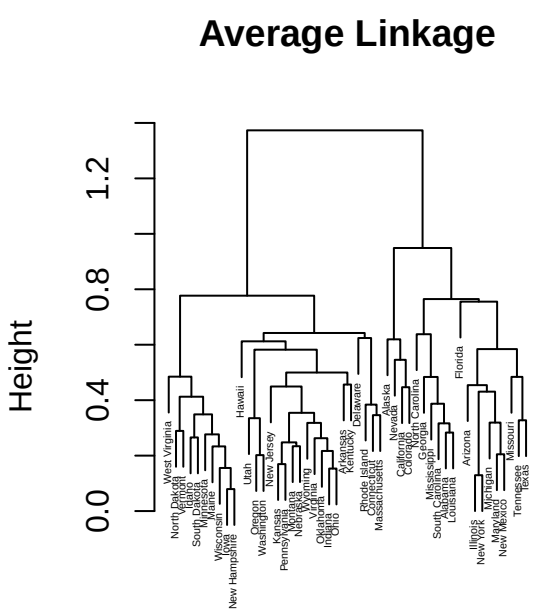
We shall use 5 clusters for k-means, based on the highest average silhoutte width.

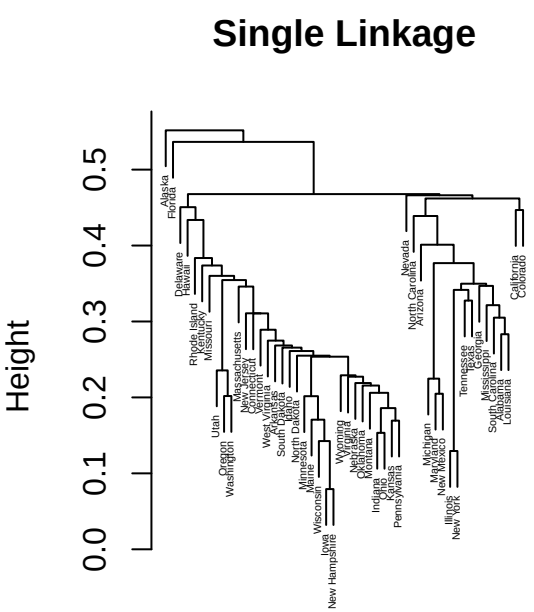
Now, we shall use hierarchical clustering. From the dendrogram below, it appears that the complete linkage does a good job in clustering our data:

```
set.seed(seed)
plot(hclust(data.dist), main = "Complete Linkage", labels = USArrests$labs, cex = 0.7)
```

Just to be sure, we try other ways:

```
par(mfrow = c(1,2))
plot(hclust(data.dist, method = "average"), main = "Average Linkage", labels = USArrests$labs, cex = 0.3
plot(hclust(data.dist, method = "single"), main = "Single Linkage", labels = USArrests$labs, cex = 0.3)
```





As expected, single linkage is the worst of all. The average linkage is okay, but it appears that complete linkage is the best. We check the distribution of different cuts:

```
for (i in 3:15){
  set.seed(seed)
  hc.out <- hclust(data.dist)
  hc.clusters <- cutree(hc.out, i)
  table(hc.clusters) %>%  print()
}
```

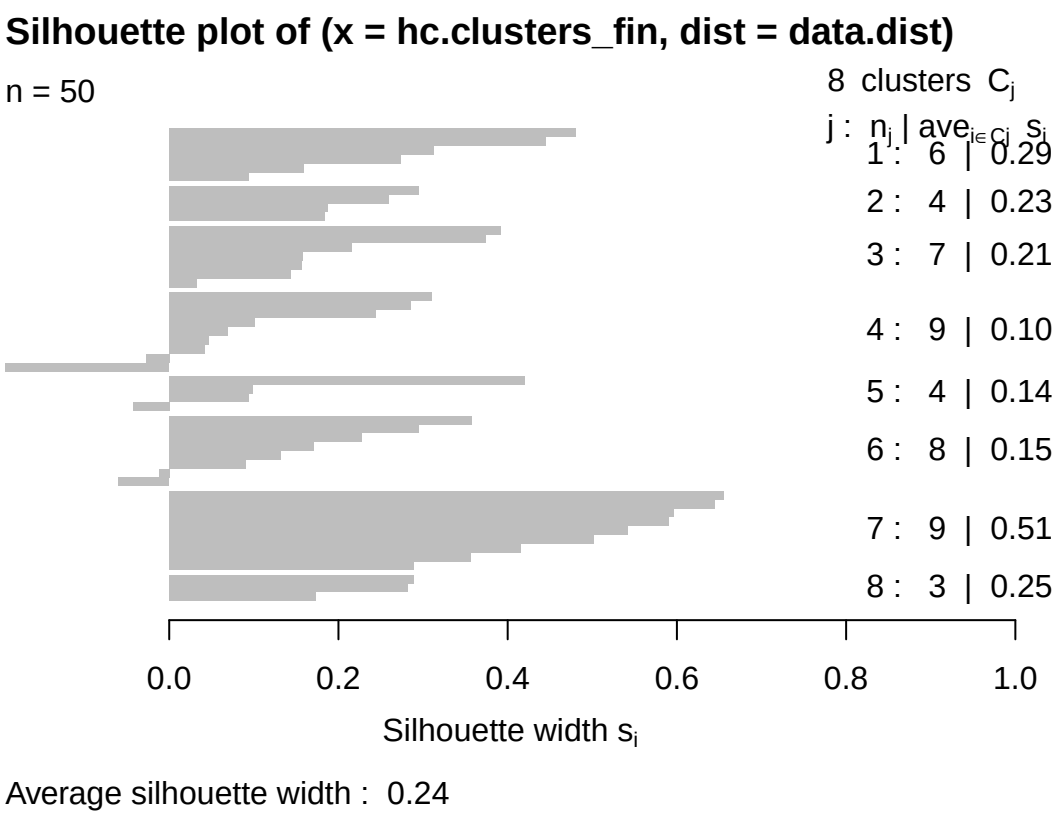
```
## hc.clusters
##  1  2  3
##  6 14 30
## hc.clusters
##  1  2  3  4
##  6 14 21  9
## hc.clusters
##  1  2  3  4  5
##  6  7  7 21  9
## hc.clusters
##  1  2  3  4  5  6
##  6  4  7 21  9  3
## hc.clusters
##  1  2  3  4  5  6  7
##  6  4  7 13  8  9  3
## hc.clusters
## 1 2 3 4 5 6 7 8
```

```
## 6 4 7 9 4 8 9 3
## hc.clusters
## 1 2 3 4 5 6 7 8 9
## 5 4 7 9 4 8 9 3 1
## hc.clusters
##  1  2  3  4  5  6  7  8  9 10
##  5  4  6  9  4  1  8  9  3  1
## hc.clusters
##  1  2  3  4  5  6  7  8  9 10 11
##  5  4  6  4  4  1  8  9  5  3  1
## hc.clusters
##  1  2  3  4  5  6  7  8  9 10 11 12
##  5  4  6  4  3  1  1  8  9  5  3  1
## hc.clusters
##  1  2  3  4  5  6  7  8  9 10 11 12 13
##  5  4  6  4  3  1  1  1  9  7  5  3  1
## hc.clusters
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14
##  5  2  6  4  2  3  1  1  1  9  7  5  3  1
## hc.clusters
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
##  5  2  6  4  2  3  1  1  1  9  4  5  3  1  3
```

Again, clusters 8 and 9 appears to be well-distributed.

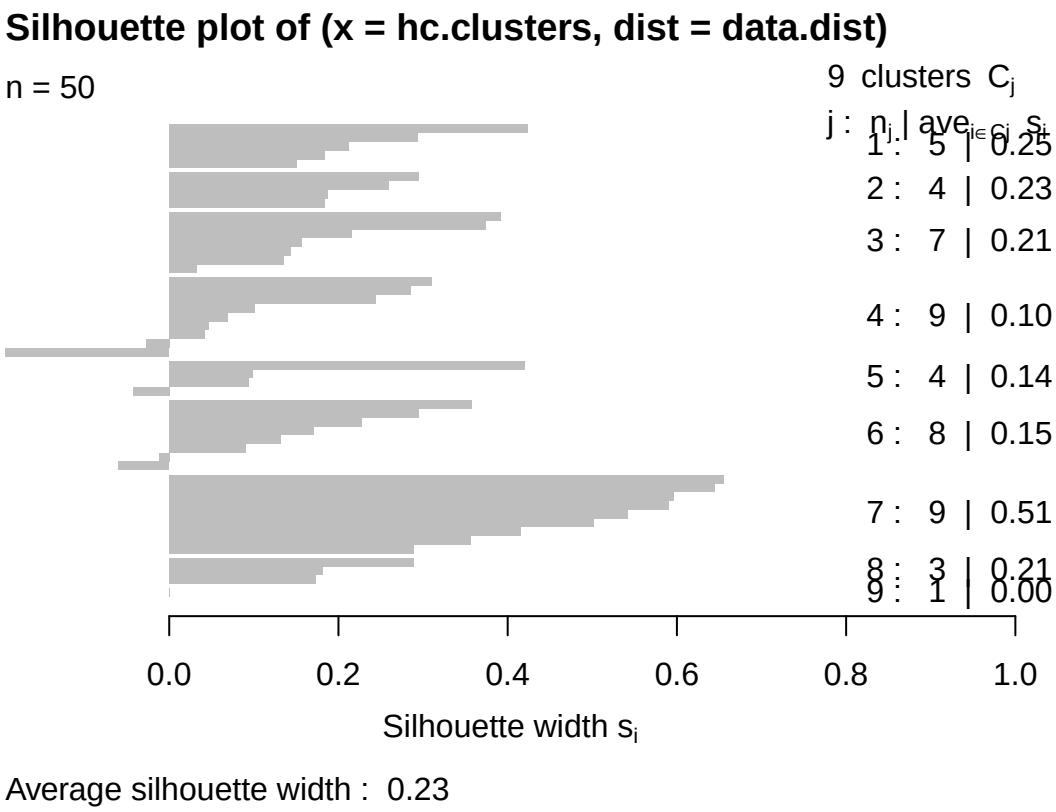
We check the silhoutte plot for 8:

```
set.seed(seed)
hc.clusters_fin <- cutree(hc.out, 8)
plot(silhouette(hc.clusters_fin, data.dist))
```



And for 9:

```
set.seed(seed)
hc.clusters <- cutree(hc.out, 9)
plot(silhouette(hc.clusters, data.dist))
```



Based on the silhoutte plot average width, we will choose 8.

For this exercise, we will not use principal component analysis as there is enough number of observations compared to the variables.

2) Provide a comparison of the clusters that you have formed - what makes them different from one another? Provide supporting summary measures and/or plots.

We first compare the distribution of the clusters:

```
table(km.out_fin$cluster)
```

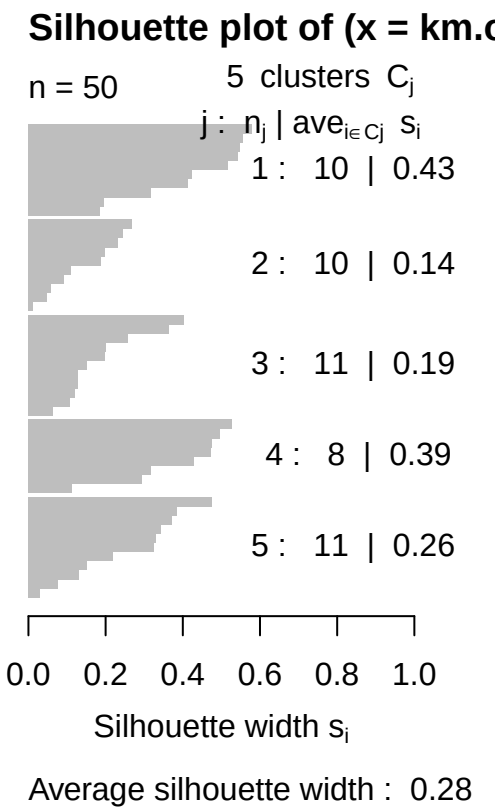
```
##
##  1  2  3  4  5
## 10 10 11  8 11
```

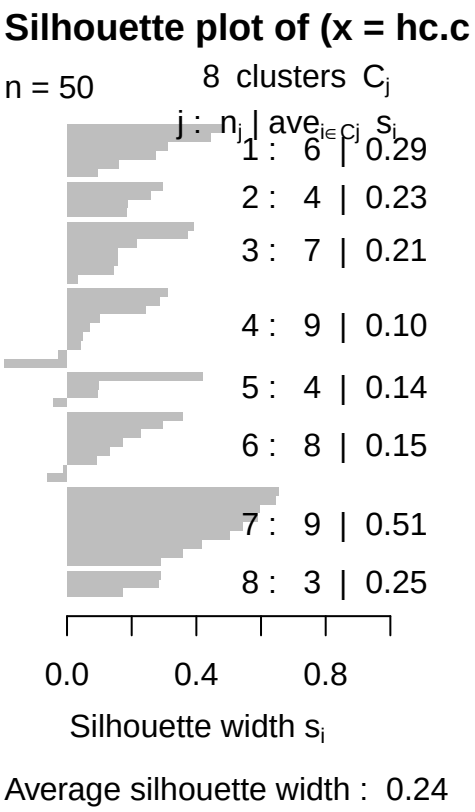
```
table(hc.clusters_fin)
```

```
## hc.clusters_fin
## 1 2 3 4 5 6 7 8
## 6 4 7 9 4 8 9 3
```

For both the clusters, we can say that both are quite well-distributed. Based on the silhoutte plot, we compare:

```
par(mfrow = c(1,2))
plot(silhouette(km.out_fin$cluster, data.dist))
plot(silhouette(hc.clusters_fin, data.dist))
```





From this, we will choose the k-means clustering method with 5 clusters as it has the higher average silhouette width and with only two negative silhoutte widths.