Pset 4, Machine Learning

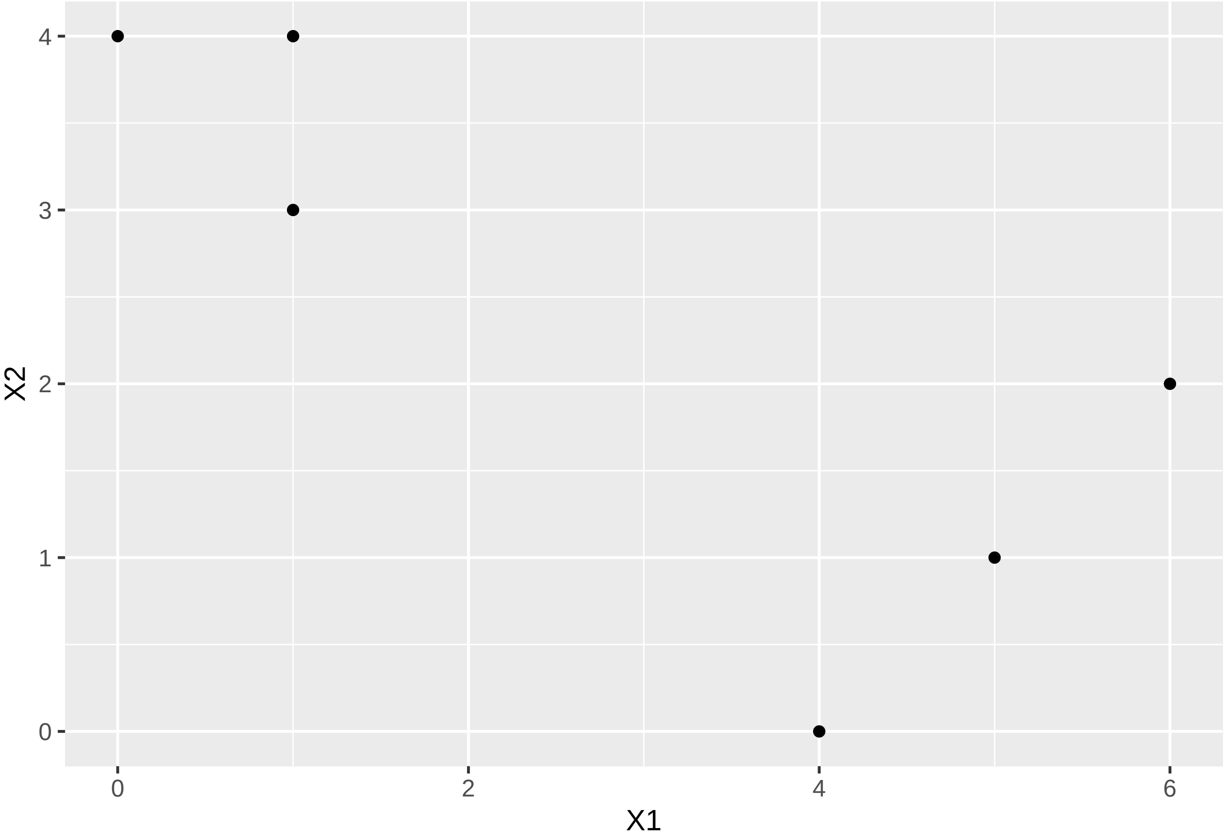
Rohen Shah

3/1/2020

Performing k-Means By Hand

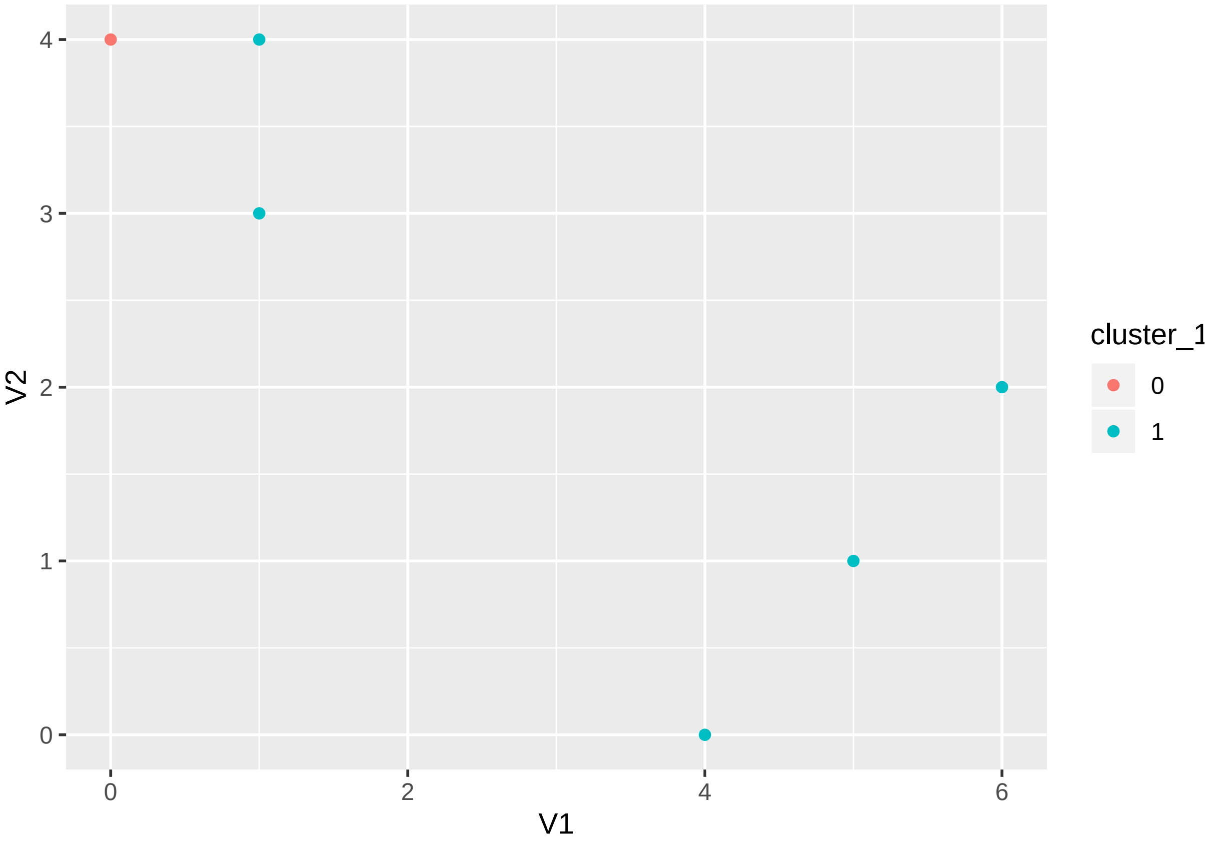
Question 1

```
x <- cbind(c(1, 1, 0, 5, 6, 4), c(4, 3, 4, 1, 2, 0))
data.frame(x) %>% ggplot(.) + geom_point(aes(x = X1, y = X2))
```



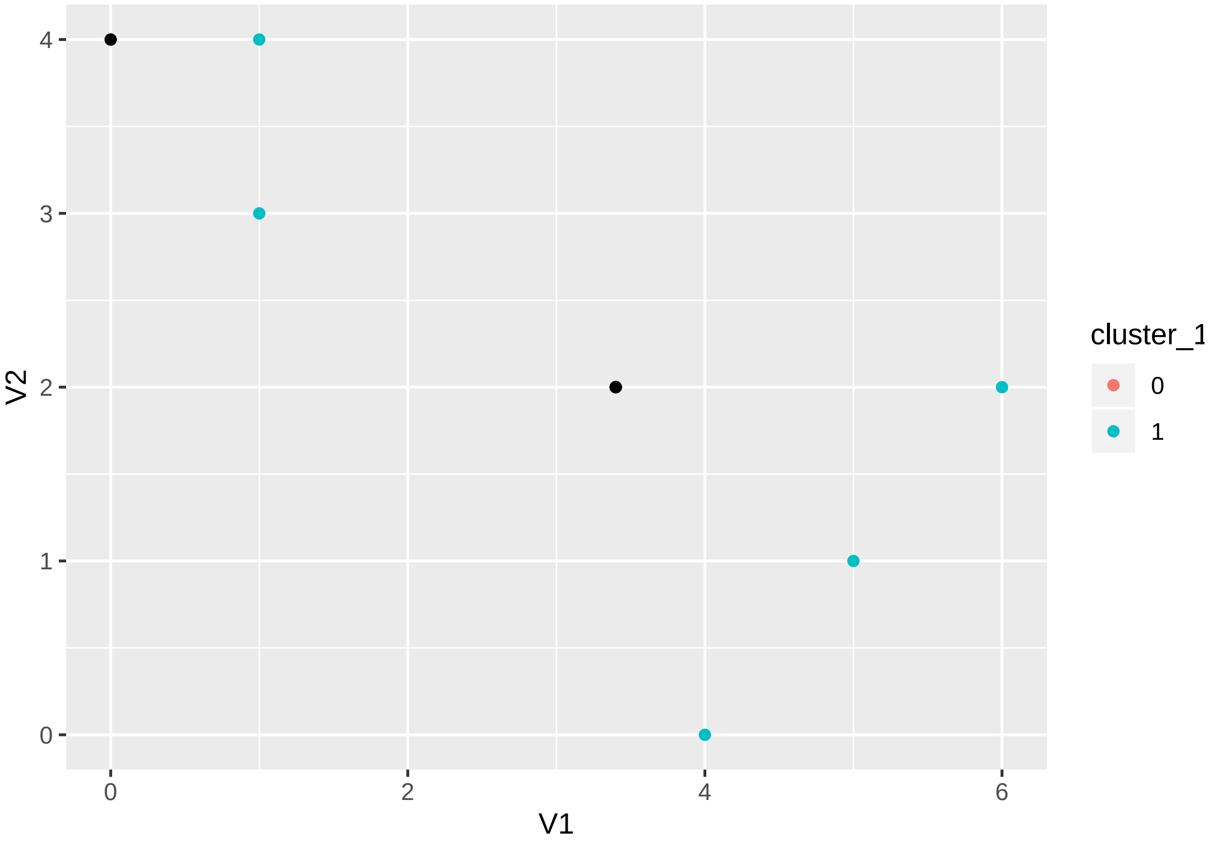
Question 2

```
set.seed(42)
n =6
cluster_1 = rbinom(n, 1, .5)
x = cbind(x, cluster_1)
data.frame(x) %>% mutate(cluster_1 = as.character(cluster_1)) %>% ggplot(.) +
        geom_point(aes(x = V1, y = V2, color = cluster_1))
```



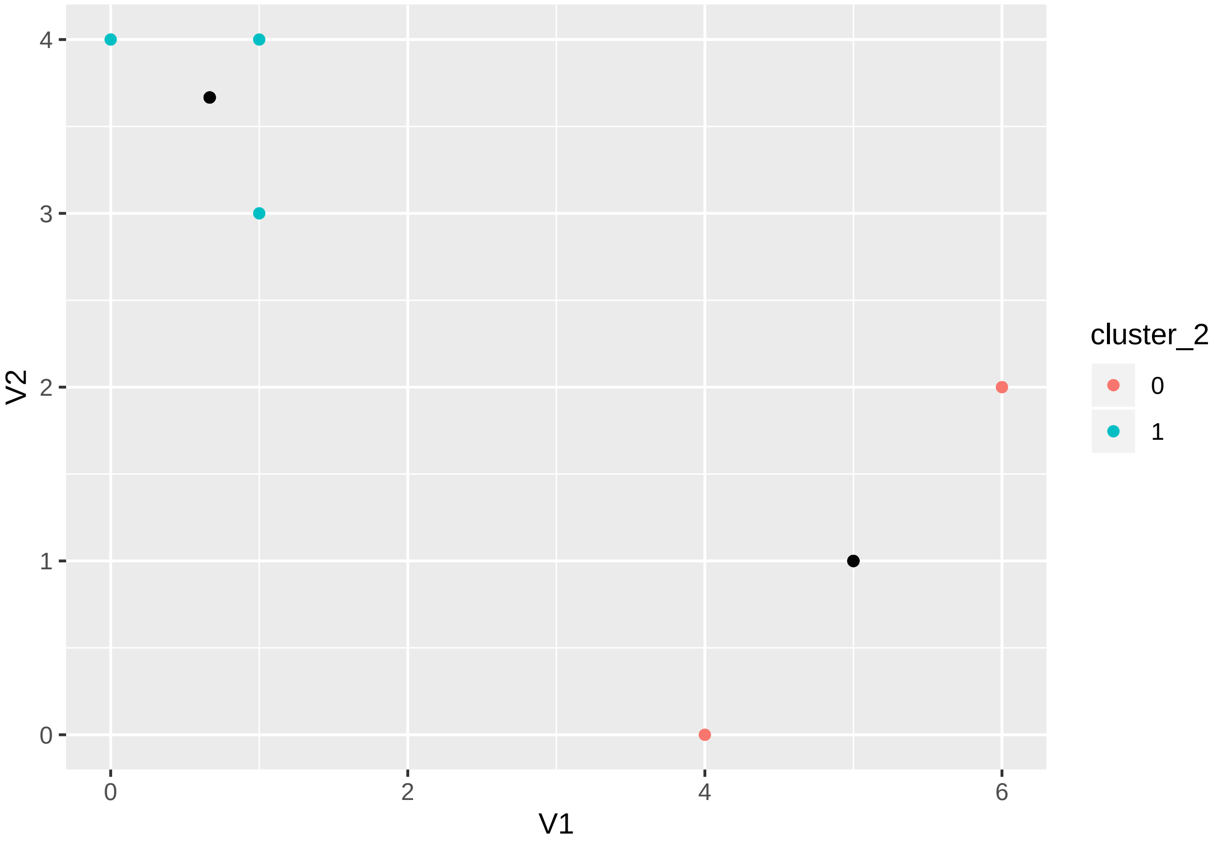
Question 3

```
x <-data.frame(x) %>% group_by(cluster_1) %>% mutate(c_x = sum(V1)/n(), c_y = sum(V2)/n()) %>% ungroup(
x %>% mutate(cluster_1 = as.character(cluster_1)) %>%
  ggplot() + geom_point(aes(x = V1, y = V2, color = cluster_1)) + geom_point(aes(x = c_x, y = c_y))
```



Question 4

```
centroids = x %>% distinct(c_x, c_y)
x_1 = tibble()
for(i in 1:6) {
  row = x[i, ]
  c1 = sqrt((row$V1 - centroids$c_x[1])^2 + (row$V2 - centroids$c_y[1])^2)
  c2 = sqrt((row$V1 - centroids$c_x[2])^2 + (row$V2 - centroids$c_y[2])^2)
  df = row %>% mutate(cluster_2 = ifelse(c1 > c2, 1, 0))
  x_1 = bind_rows(x_1, df)
}
x_1 <- x_1 %>% group_by(cluster_2) %>% mutate(c_x = sum(V1)/n(), c_y = sum(V2)/n()) %>% ungroup()
x_1 %>% mutate(cluster_2 = as.character(cluster_2)) %>%
    ggplot() + geom_point(aes(x = V1, y = V2, color = cluster_2)) + geom_point(aes(x = c_x, y = c_y))
```

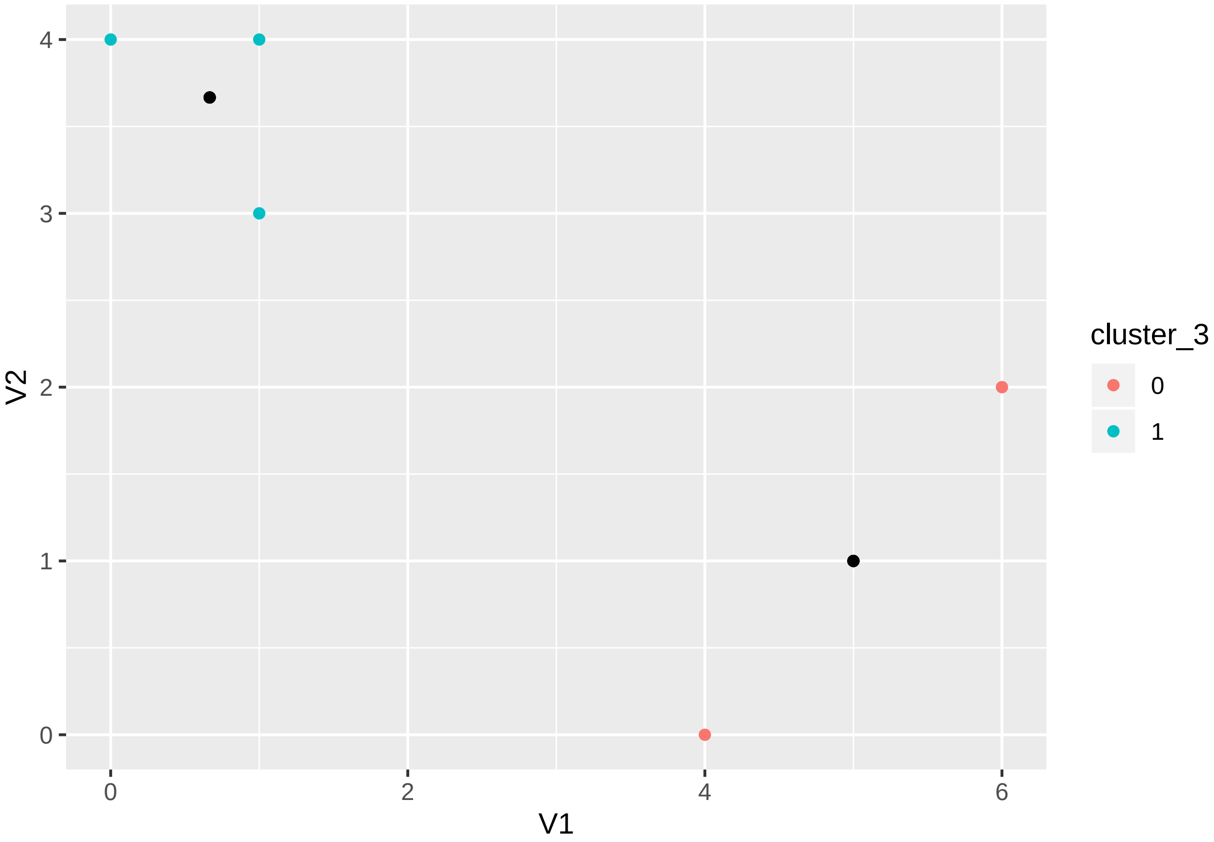


Question 5

As we can see, I got lucky where the current assignments (with the seed I chose) are already the closest (cluster one being the top left 3 points, cluster 0 being the bottom 3 points), so when I run the code again to re-run the clustering, it's the same outcome as before.

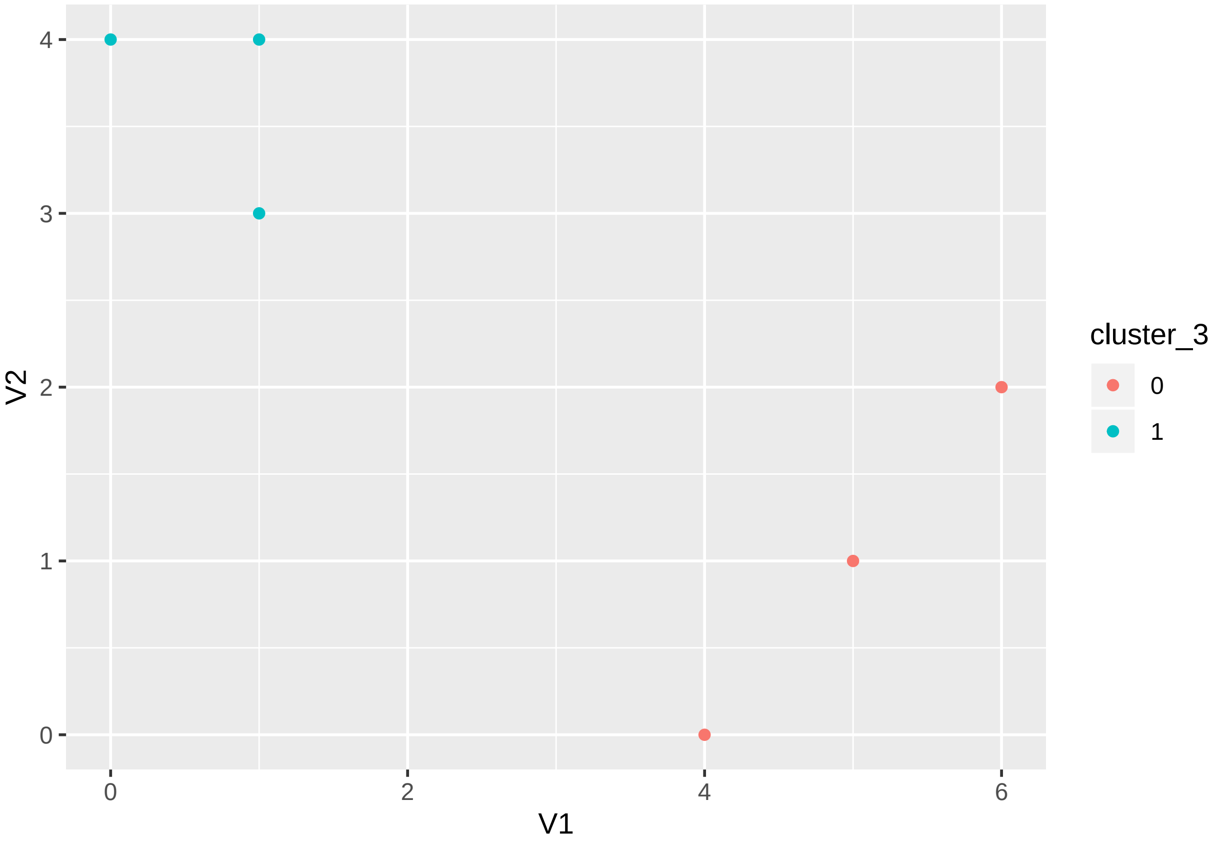
```
centroids_2 <- x_1 %>% distinct(c_x, c_y)
x_2 = tibble()
for(i in 1:6) {
  row = x_1[i, ]
  c1 = sqrt((row$V1 - centroids_2$c_x[1])^2 +(row$V2 - centroids_2$c_y[1])^2)
  c2 = sqrt((row$V1 - centroids_2$c_x[2])^2 +(row$V2 - centroids_2$c_y[2])^2)
  df = row %>%
  mutate(cluster_3 = ifelse(c1 < c2, 1, 0))
x_2 = bind_rows(x_2, df) }

x_2 %>% mutate(cluster_3 = as.character(cluster_3)) %>%
  ggplot() + geom_point(aes(x = V1, y = V2, color = cluster_3)) + geom_point(aes(x = c_x, y = c_y))
```



Question 6

```
x_2 %>% mutate(cluster_3 = as.character(cluster_3)) %>%
  ggplot() + geom_point(aes(x = V1, y = V2, color = cluster_3))
```



Clustering State Legislative Professionalism

Question 1

```
load("~/Dropbox/2. 2020 Winter/PLSC 43505/Problem Sets/Pset 4/legprof-components.v1.0.RData")
```

Question 2

```
x=x[,c("state", "year", "t_slength", "slength", "salary_real", "expend")]
x = x %>% filter(year == 2009 | year == 2010) %>% na.omit() %>% data.frame()
```

Question 3

This helps us get the distance, which will be used later on.

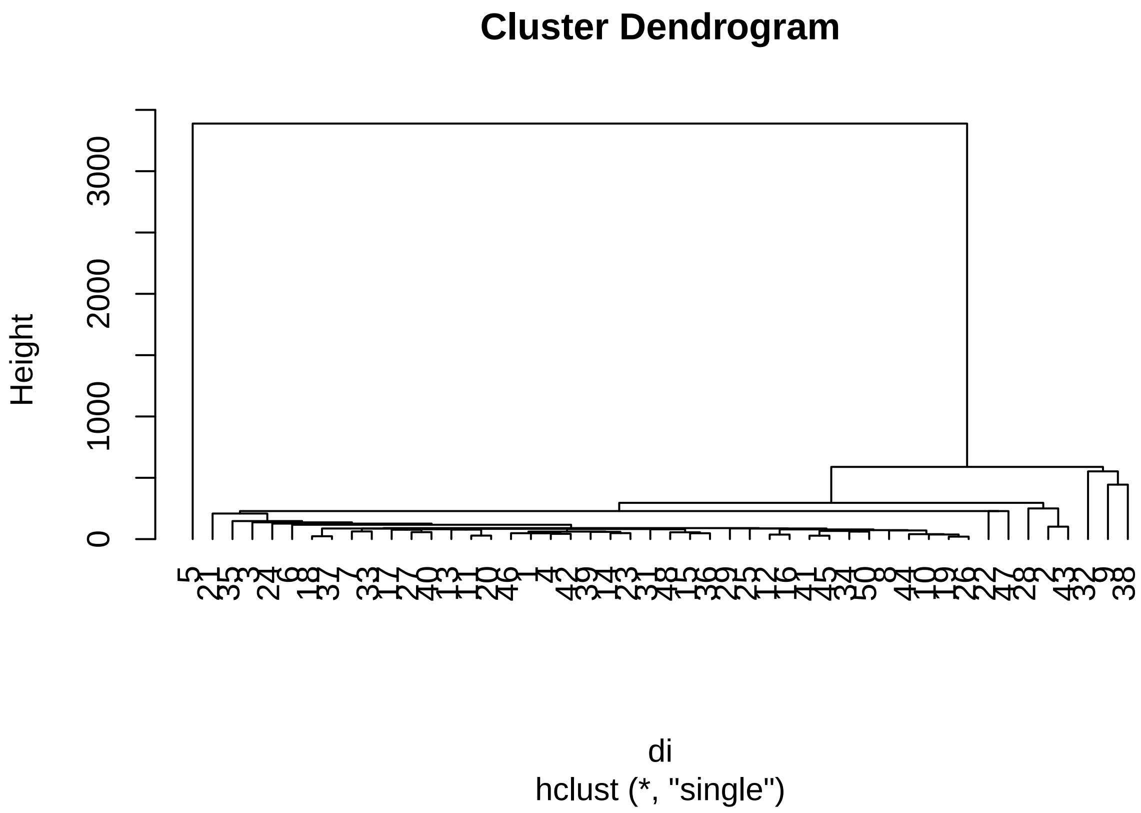
```
di = get_dist(x)
```

```
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion
```

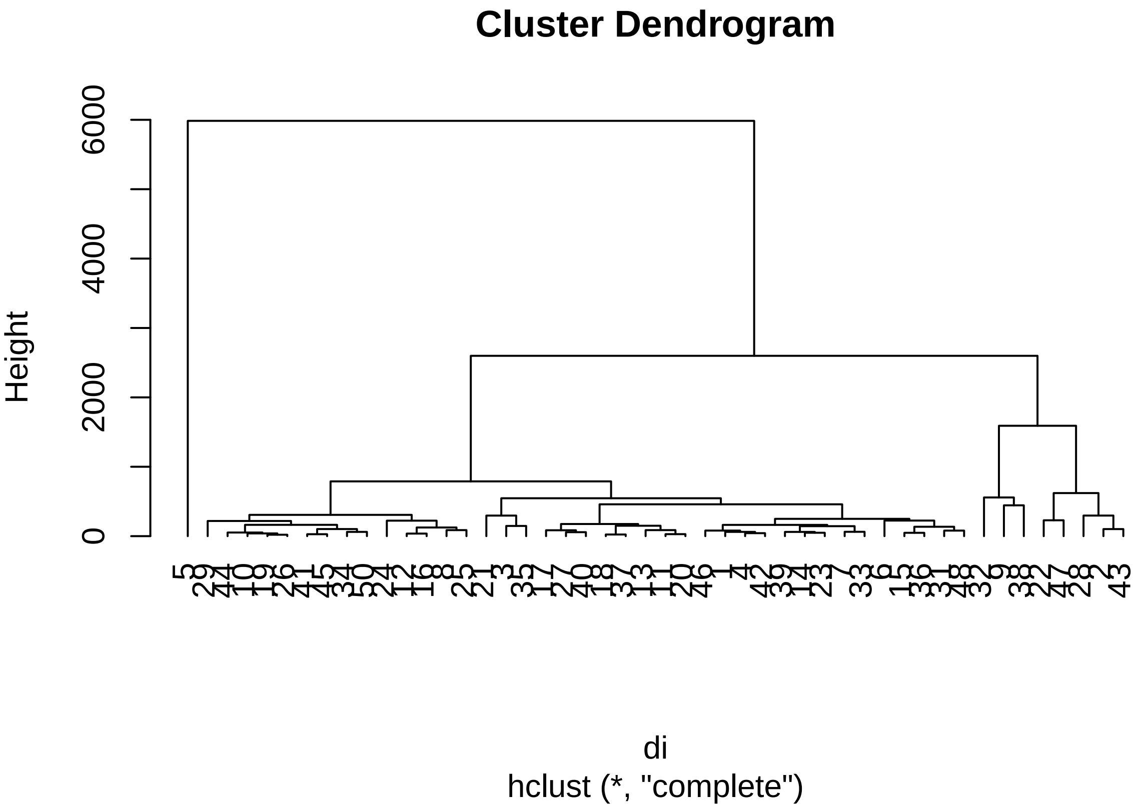
Question 4

The HAC algorithm shows that there are primary clusters; one with a few large states and the other with many smaller ones.

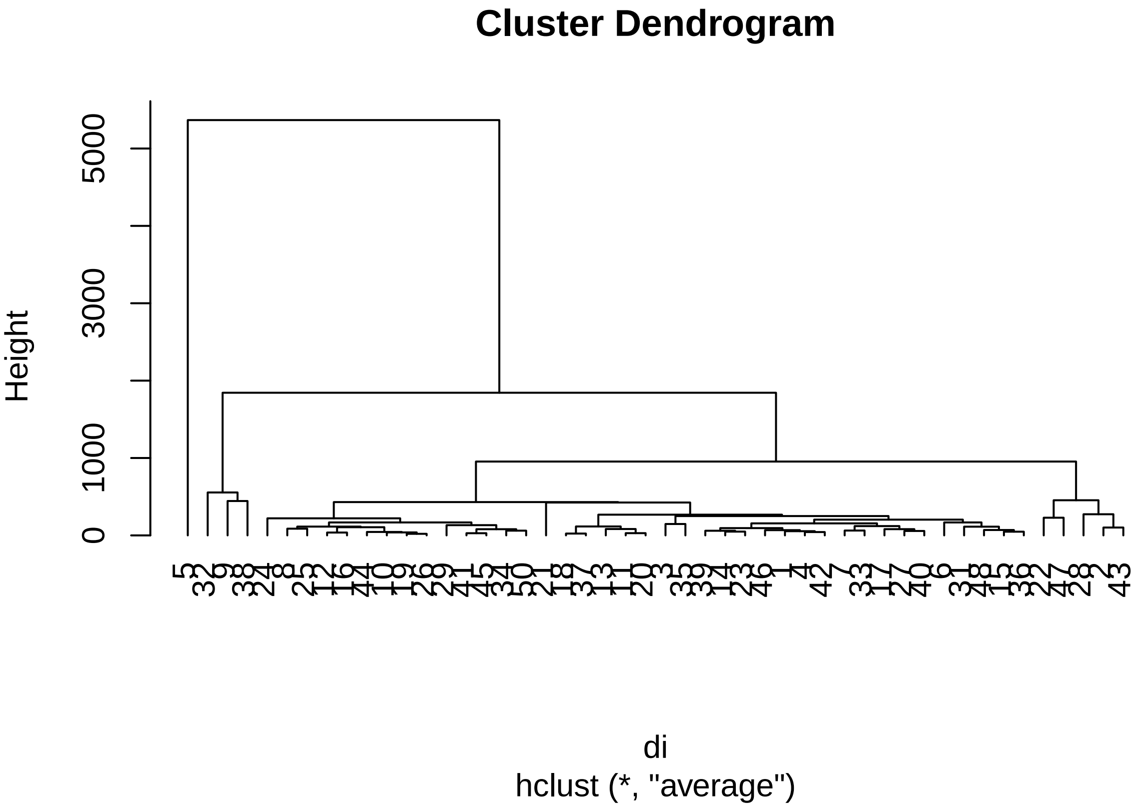
```
hc_single <- hclust(di,
                    method = "single");plot(hc_single, hang = -1)
```



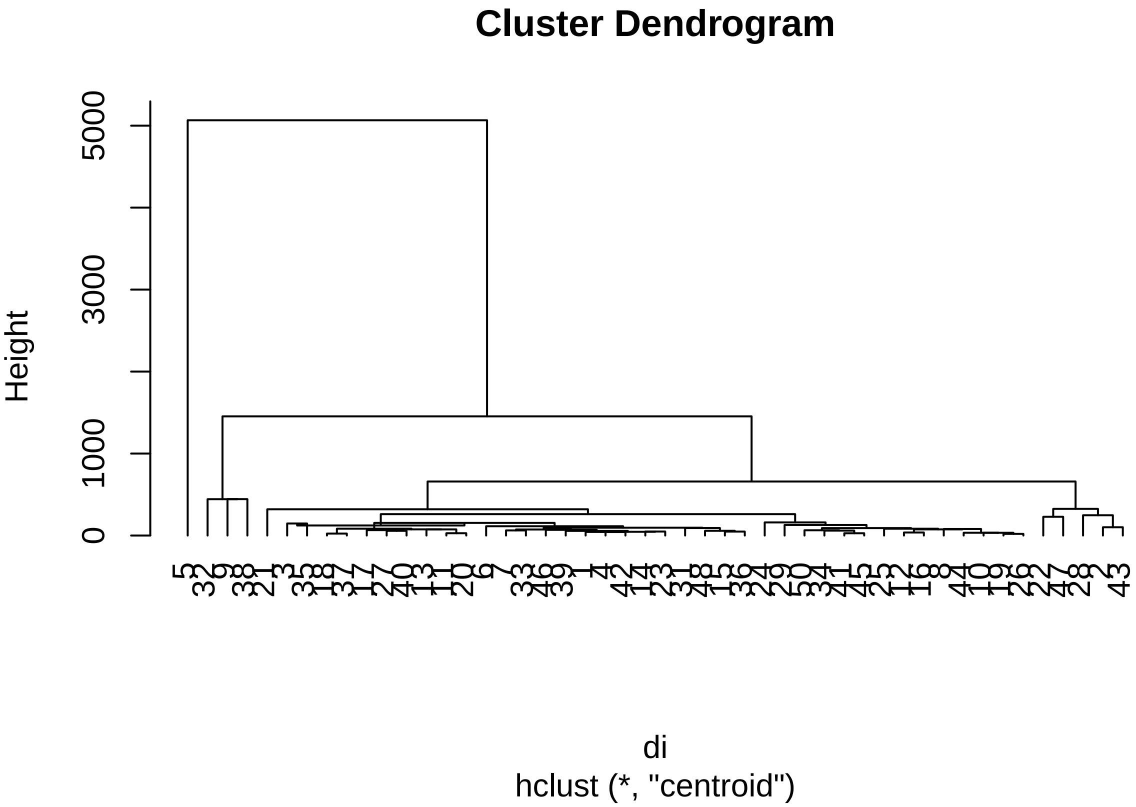
```
hc_complete <- hclust(di,
                      method = "complete"); plot(hc_complete, hang = -1)
```



```
hc_average <- hclust(di,
                     method = "average"); plot(hc_average, hang = -1)
```



```r
hc_centroid <- hclust(di,
                      method = "centroid"); plot(hc_centroid, hang = -1)
```



Question 5

```r
set.seed(42)
kmeans <- kmeans(x[ ,2],
                 centers = 2, #k
                 nstart = 15) #start 15 times
kmeans_results <- data.frame(kmeans$cluster)
kmeans_results$state
```

```
## NULL
```

```r
as.list(c1)
```

```
## [[1]]
## [1] 4.955356
```

```r
c2 <- kmeans_results%>%
  filter(kmeans$cluster == 2)%>%
as.list(c2)
```