Problem Set 4

Rei Bertoldi

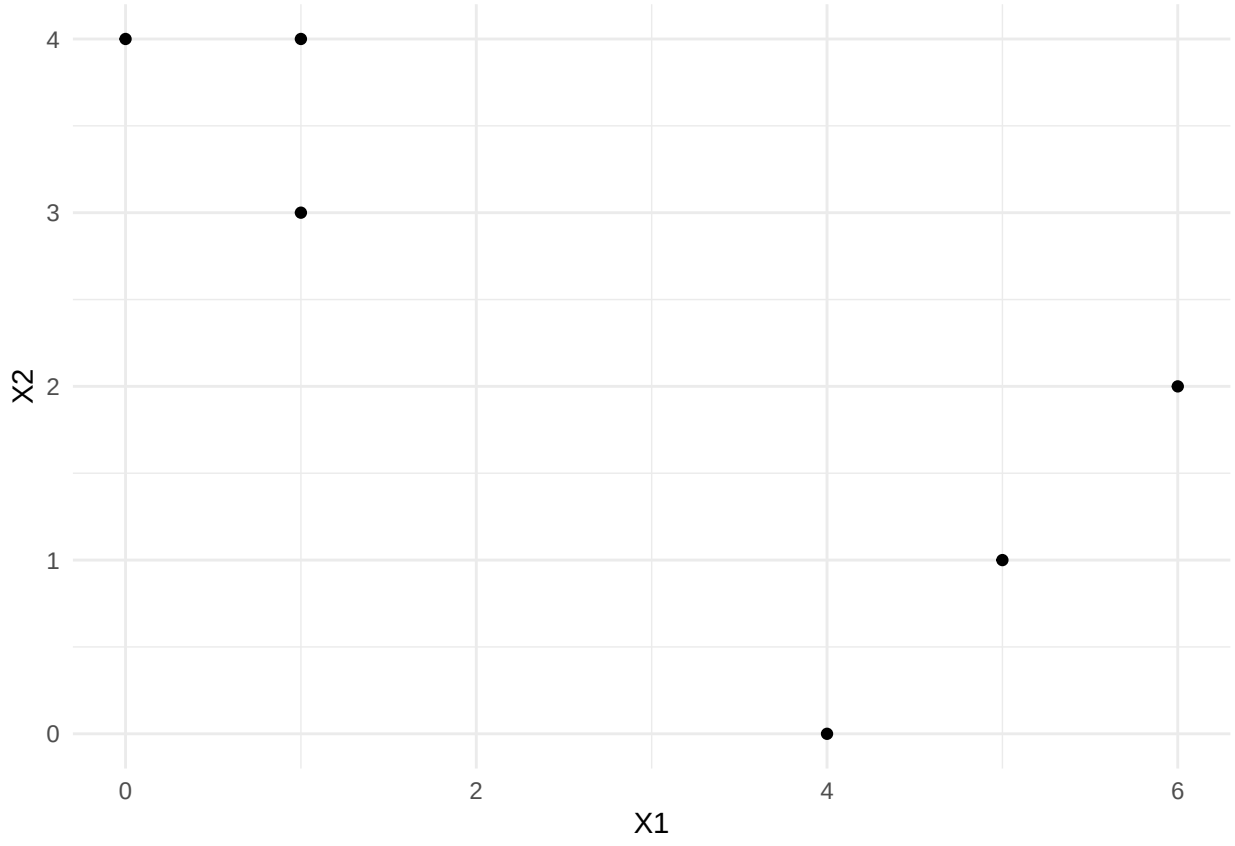
3/2/2020

Performing k-Means By Hand

```
x <- cbind(c(1, 1, 0, 5, 6, 4), c(4, 3, 4, 1, 2, 0))
```

Question 1

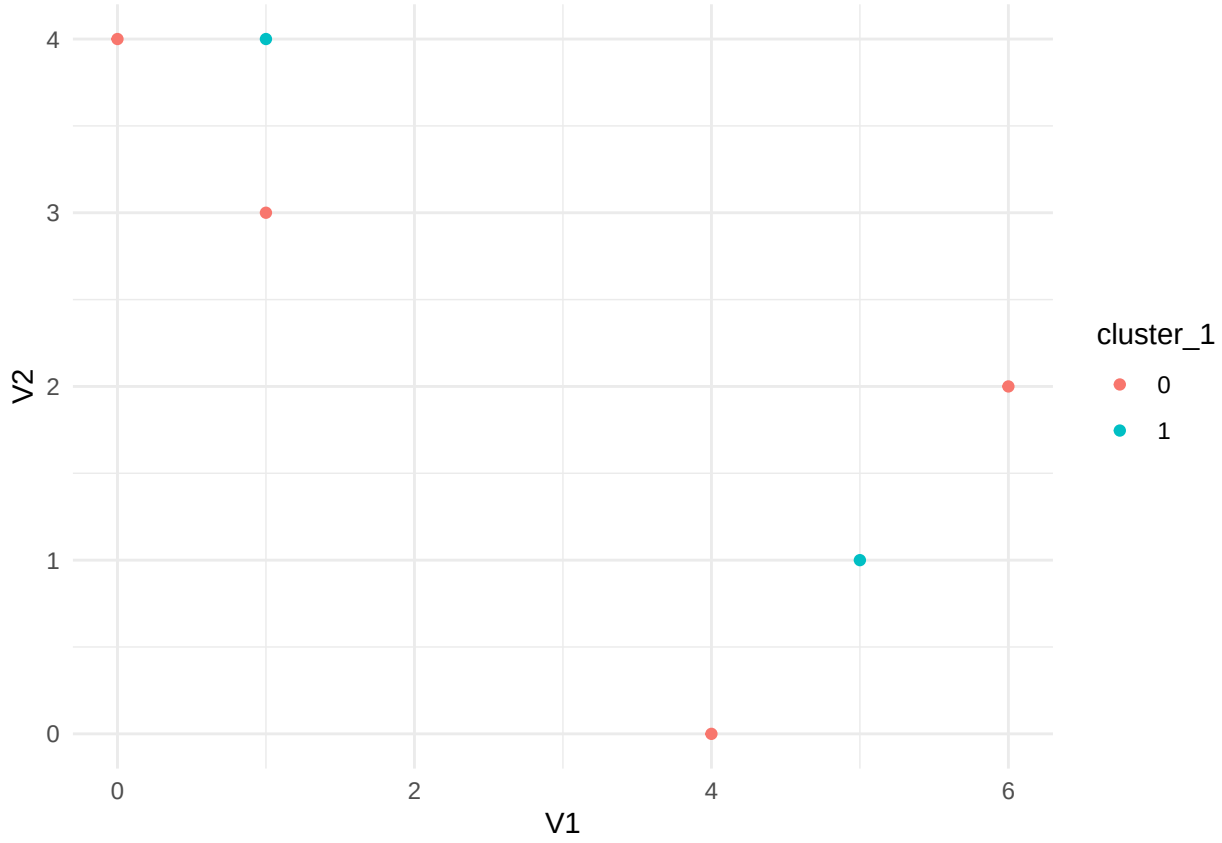
```
data.frame(x) %>%
  ggplot(.) +
  geom_point(aes(x = X1, y = X2)) +
  theme_minimal()
```



Question 2

```
set.seed(10)
theta <- 0.5
n <- 6
cluster_1 <- rbinom(n, 1, theta)
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)) +
  theme_minimal()
```

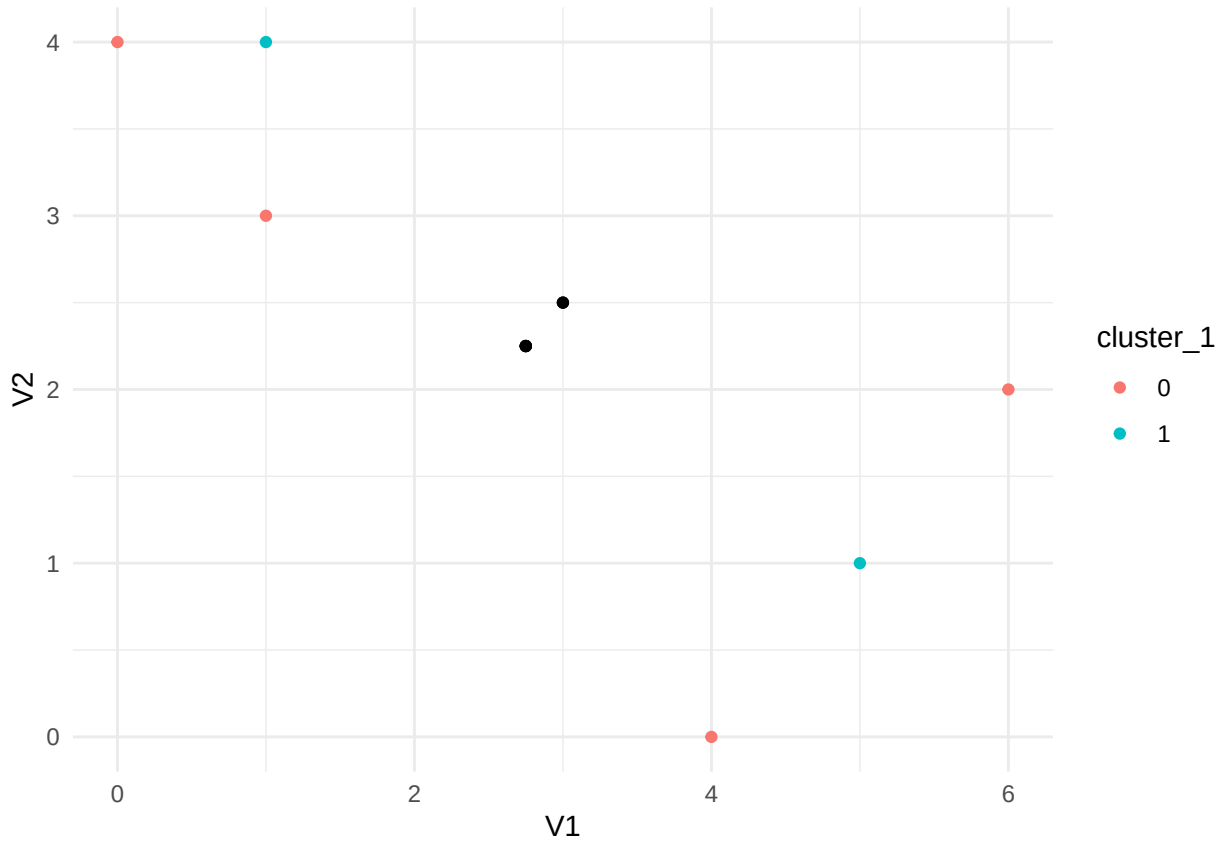


Question 3

```
x <-data.frame(x) %>%
  group_by(cluster_1) %>%
  mutate(centroid_x = sum(V1)/n(),
            centroid_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 = centroid_x, y = centroid_y)) +
  theme_minimal()
```

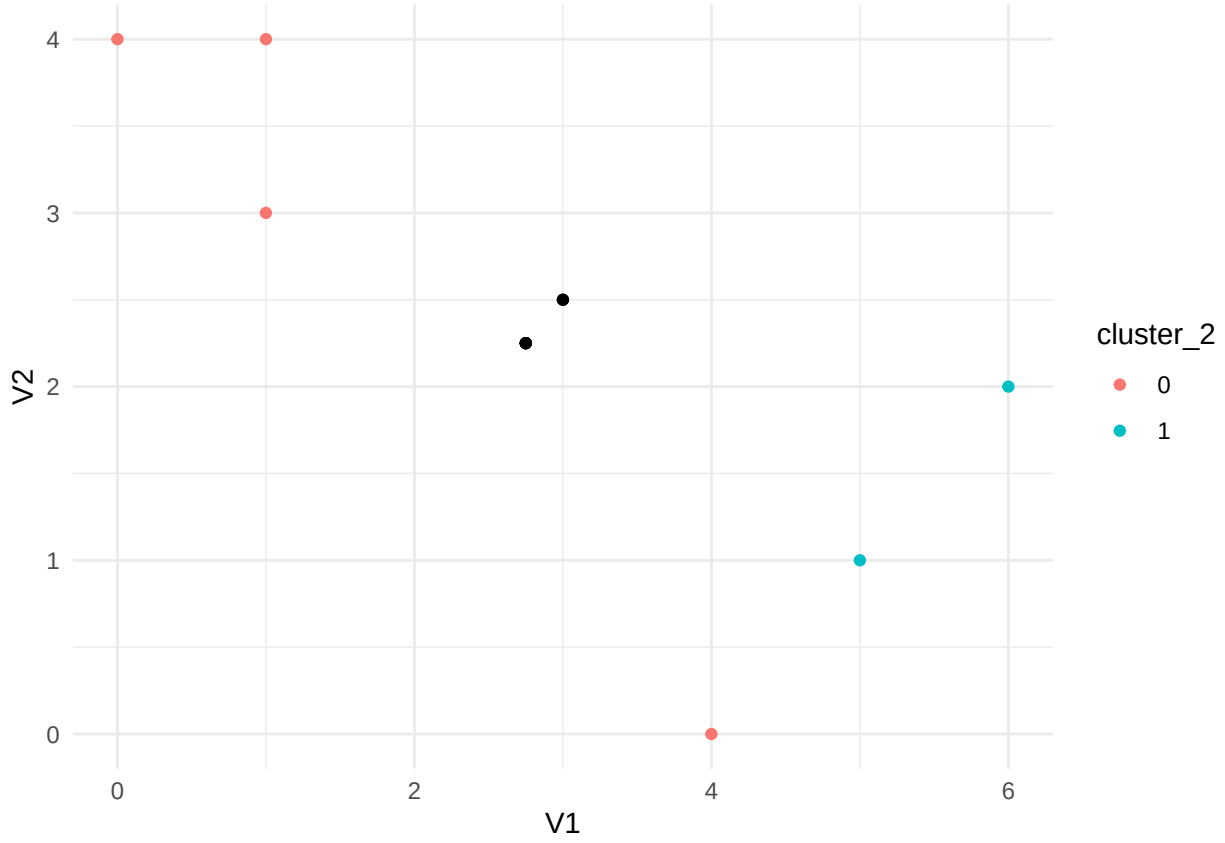


Question 4

```
centroids <- x %>%
  distinct(centroid_x, centroid_y)

x_1 <- tibble()
for(i in 1:nrow(x)) {
  row <- x[i, ]
  c1 <- sqrt((row$V1 - centroids$centroid_x[1])^2 +
               (row$V2 - centroids$centroid_y[1])^2)
  c2 <- sqrt((row$V1 - centroids$centroid_x[2])^2 +
               (row$V2 - centroids$centroid_y[2])^2)
  df <- row %>%
    mutate(cluster_2 = ifelse(c1 < c2, 1, 0))
  x_1 <- bind_rows(x_1, df)
}
```

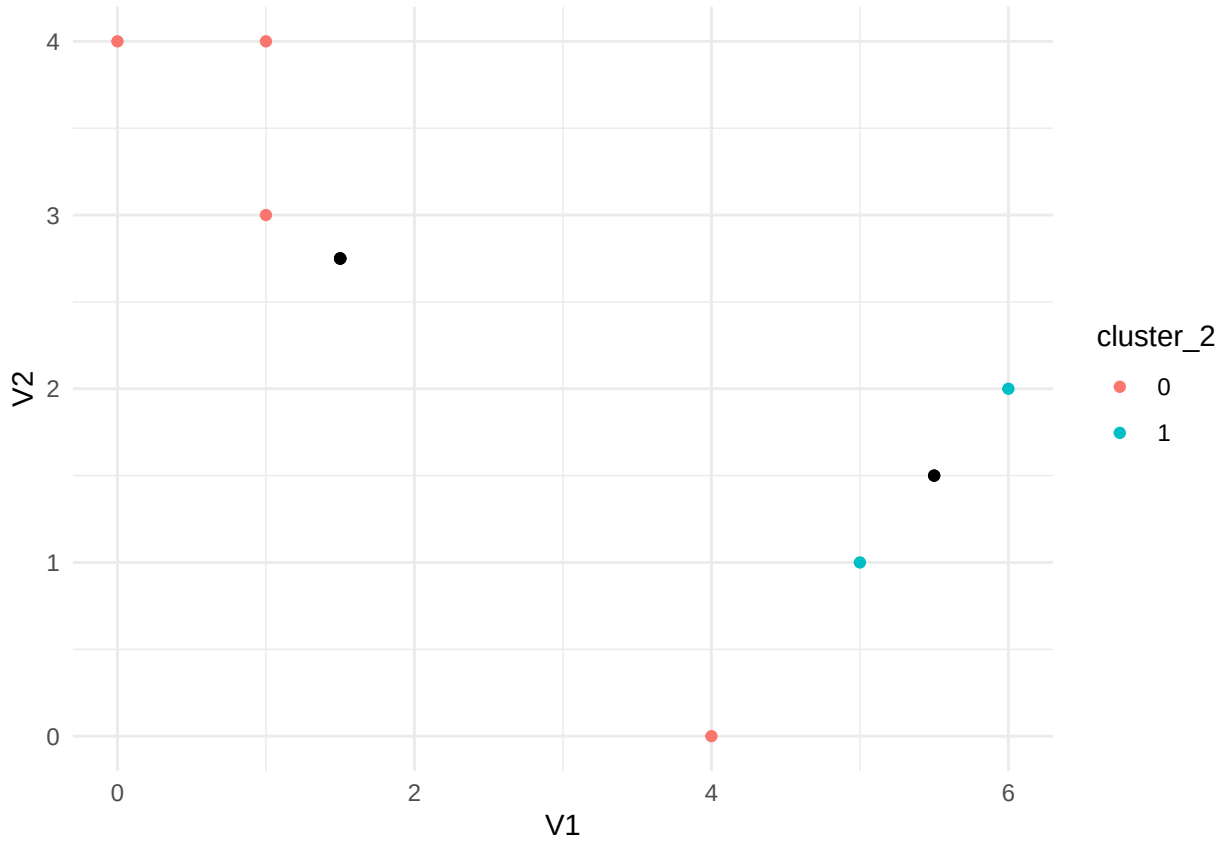
```
x_1 %>%
  mutate(cluster_2 = as.character(cluster_2)) %>%
  ggplot() +
  geom_point(aes(x = V1, y = V2, color = cluster_2)) +
  geom_point(aes(x = centroid_x, y = centroid_y)) +
```

```
  theme_minimal()
```

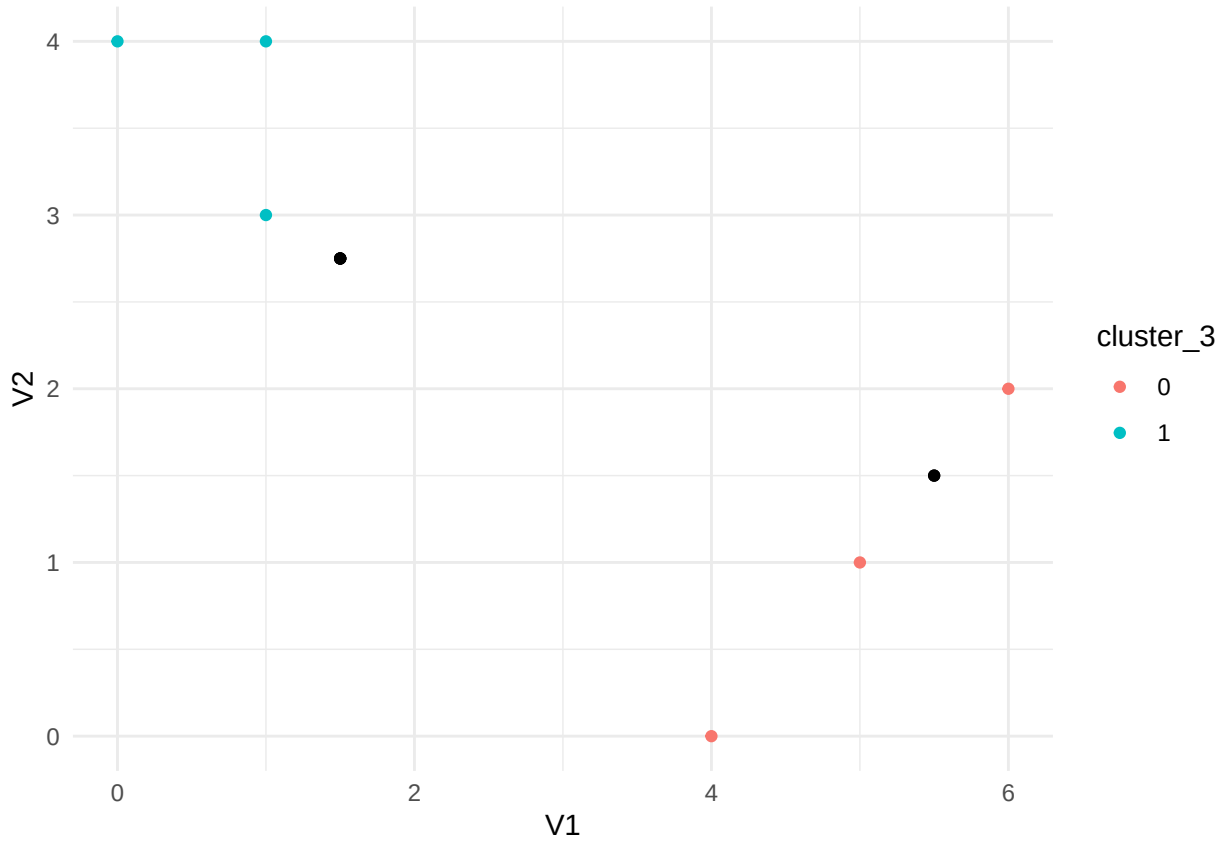


Question 5

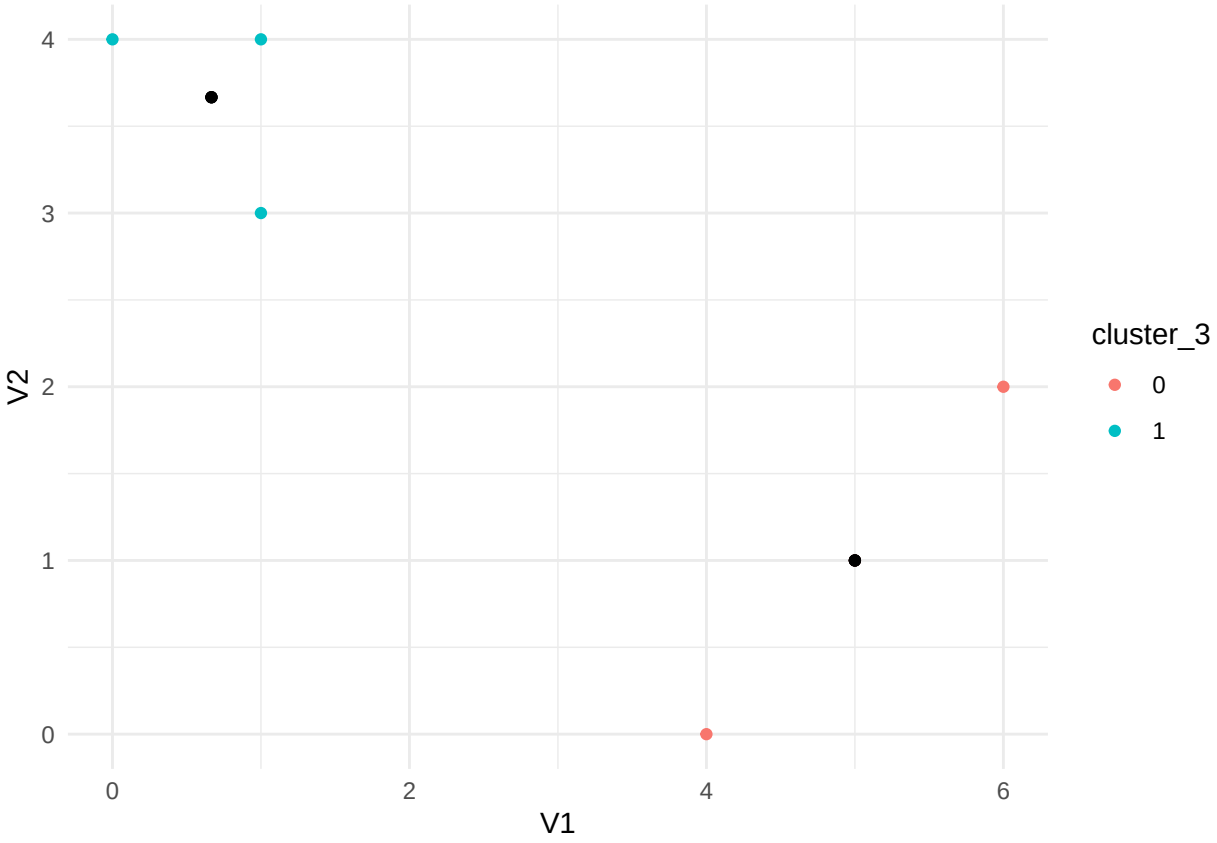
```
x_1 <- x_1 %>%
  group_by(cluster_2) %>%
  mutate(centroid_x = sum(V1)/n(),
         centroid_y = sum(V2)/n()) %>%
  ungroup()

x_1 %>%
  mutate(cluster_2 = as.character(cluster_2)) %>%
  ggplot() +
  geom_point(aes(x = V1, y = V2, color = cluster_2)) +
  geom_point(aes(x = centroid_x, y = centroid_y)) +
  theme_minimal()
```




```
centroids_2 <- x_1 %>%
  distinct(centroid_x, centroid_y)

x_2 <- tibble()
for(i in 1:nrow(x_1)) {
  row <- x_1[i, ]
  c1 <- sqrt((row$V1 - centroids_2$centroid_x[1])^2 +
               (row$V2 - centroids_2$centroid_y[1])^2)
  c2 <- sqrt((row$V1 - centroids_2$centroid_x[2])^2 +
               (row$V2 - centroids_2$centroid_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 = centroid_x, y = centroid_y)) +
  theme_minimal()
```

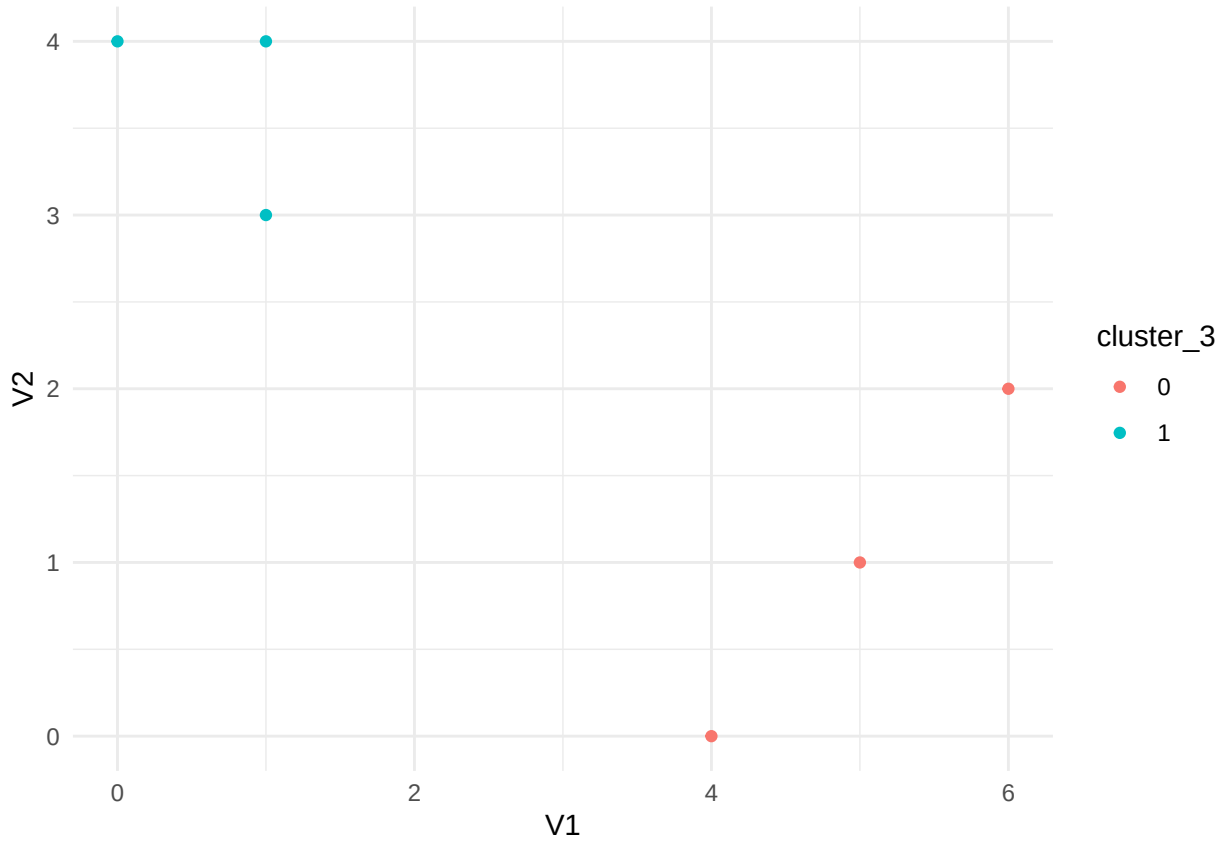



```
x_2 <- x_2 %>%
  group_by(cluster_3) %>%
  mutate(centroid_x = sum(V1)/n(),
         centroid_y = sum(V2)/n()) %>%
  ungroup()

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



Question 6

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



Clustering State Legislative Professionalism

Question 1

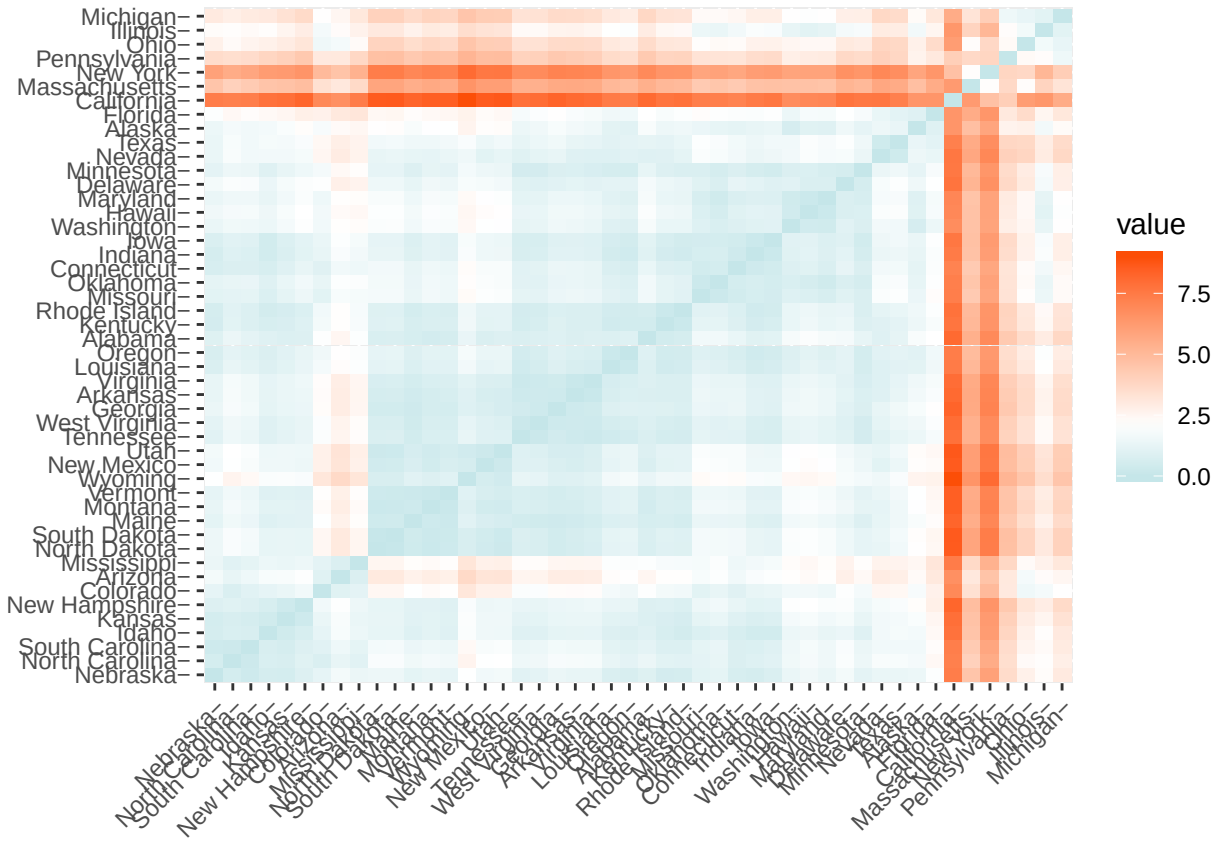
```
load("C:/Users/User02/Desktop/Harris/machine_learning/pset4/legprof-components.v1.0.RData")
```

Question 2

```
x_clean <- x %>%
  select(state, year, t_slength, slength, salary_real, expend) %>%
  filter(year == 2009 |
           year == 2010) %>%
  na.omit() %>%
  data.frame() %>%
  mutate_at(c(3:6), funs(c(scale(.)))) %>%
  remove_rownames %>%
  column_to_rownames(var = "state") %>%
  select(-year)
```

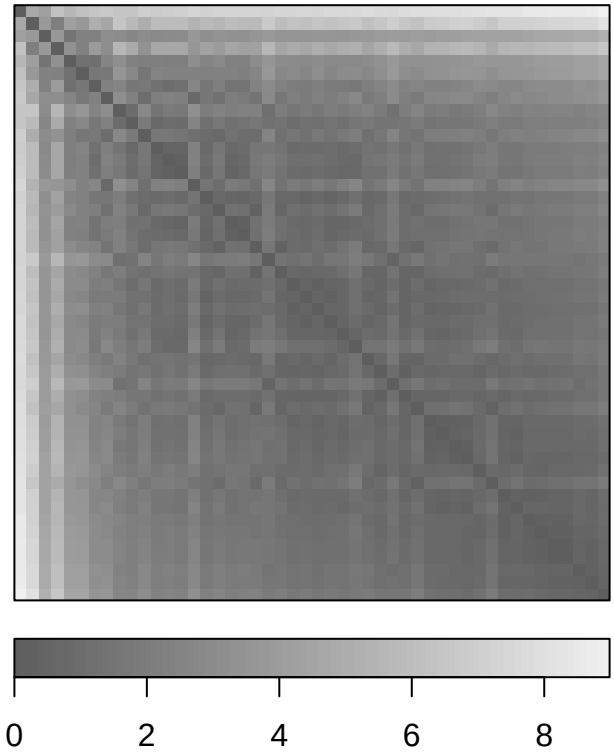
Question 3

```
distance <- get_dist(x_clean)
fviz_dist(distance,
          gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))
```



Computing and visualizing a distance matrix between rows of the data.

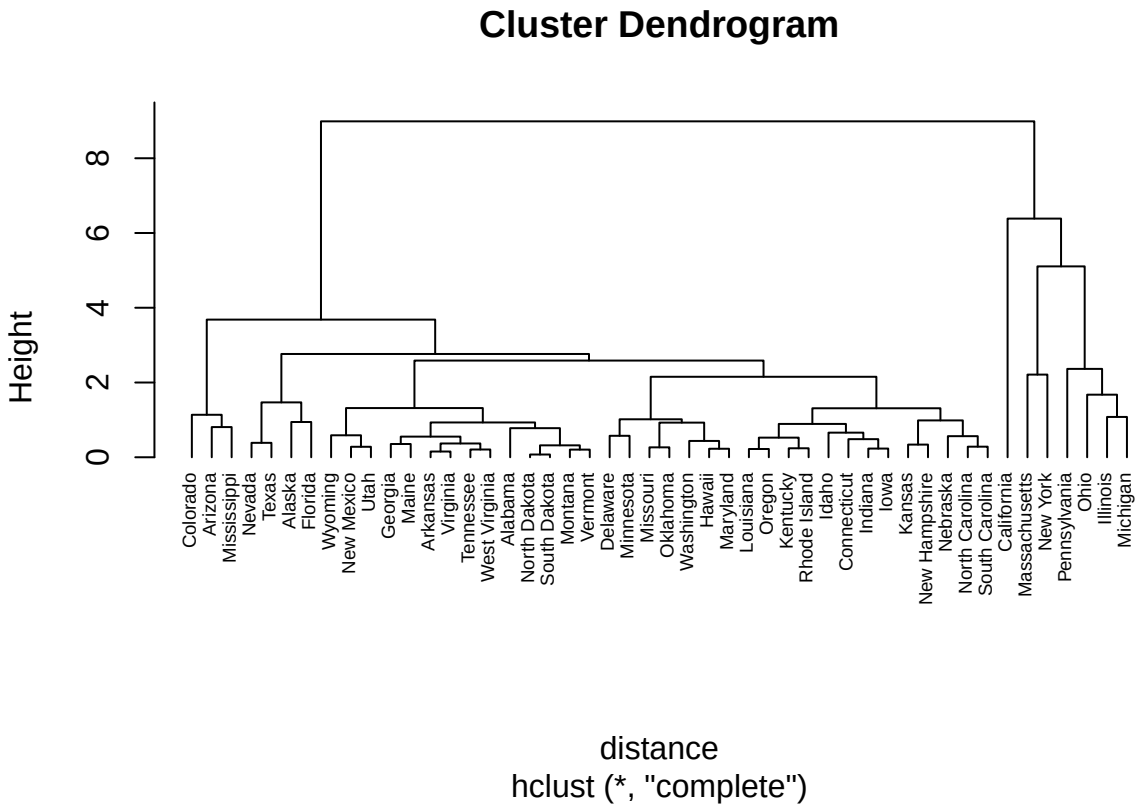
```
dissplot(distance)
```



Visualizing a dissimilarity matrix.

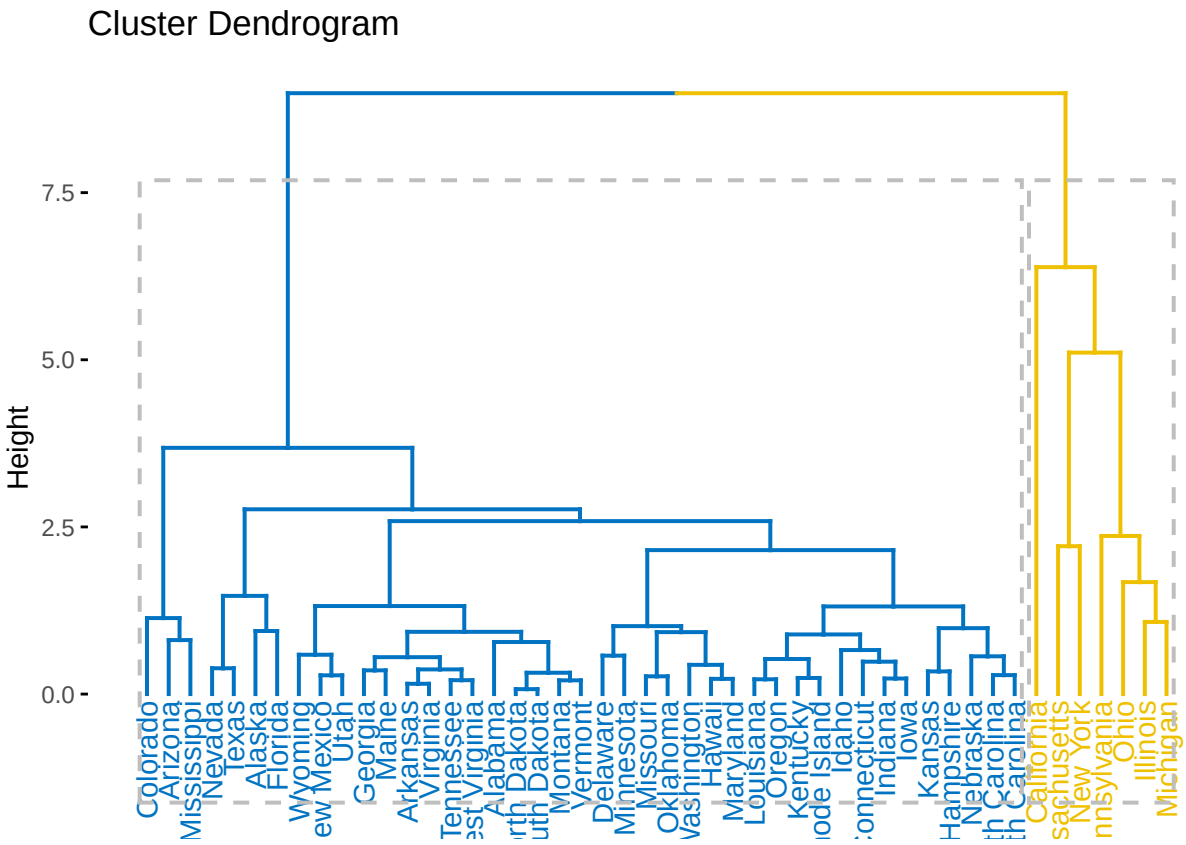
Question 4

```
hc <- hclust(distance, method = "complete")
plot(hc, cex = 0.6, hang = -1)
```



There seem to be two clear clusters. One of the clusters contains 41 of the observations and 7 in the other. Within the smaller cluster of seven states, California seems to be distict, on its own branch. California has high legistlative professonialism, so this is to be expected. The states in the smaller cluster generally have high professonialism, so these clustering results makes sense.

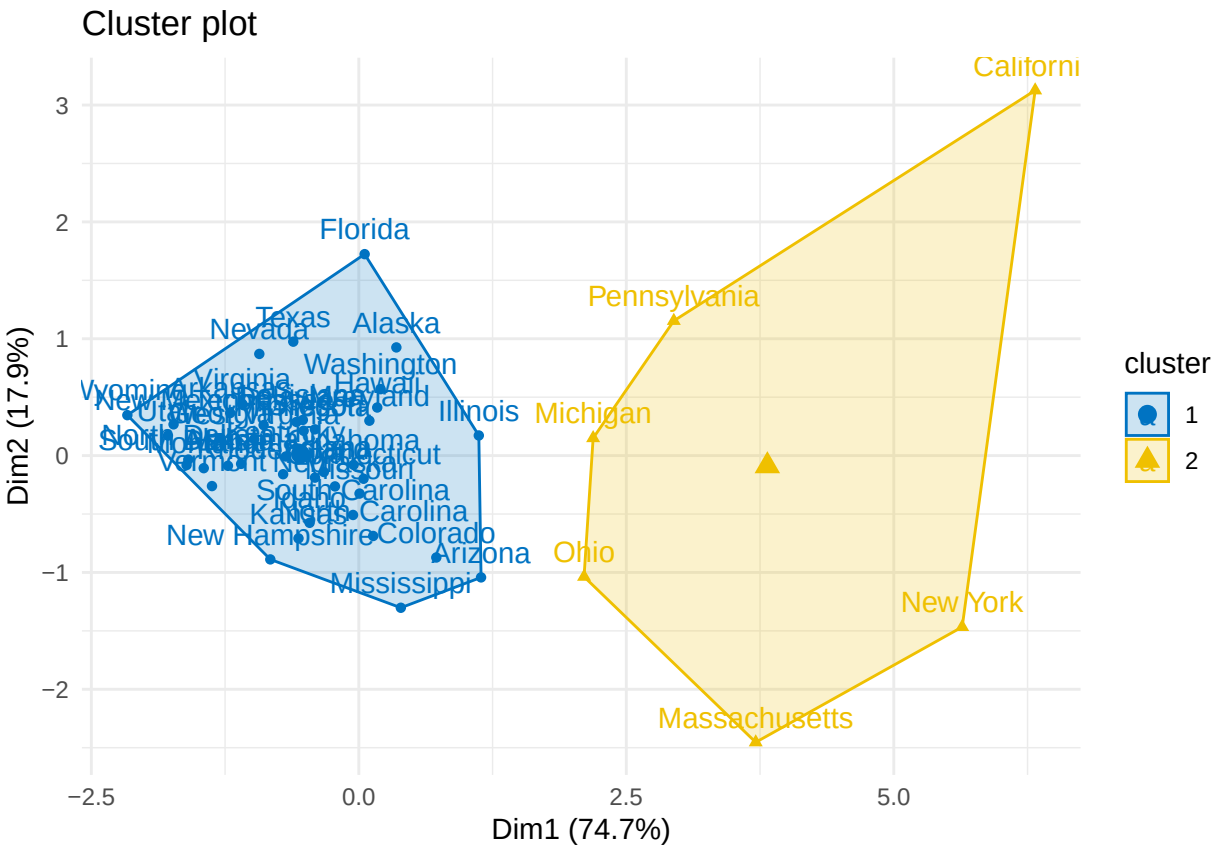
```
hc.cut <- hcut(x_clean, k = 2, hc_method = "complete")
fviz_dend(hc.cut, rect = TRUE, palette = "jco", as.ggplot = TRUE)
```



Cutting the tree, k =2. The plot clearly illustrates the two clusters.

Question 5

```
km <- kmeans(x_clean, centers = 2, nstart = 15)
fviz_cluster(km, data = x_clean, palette = "jco", as.ggplot = TRUE) +
  theme_minimal()
```



K-means cluster moved Illinois into the larger cluster, so now the larger cluster comprises 42 observations, while the smaller has 6 observations. Again, we can see from the plot that California is standing out as a state with very high professionalism.

Question 6

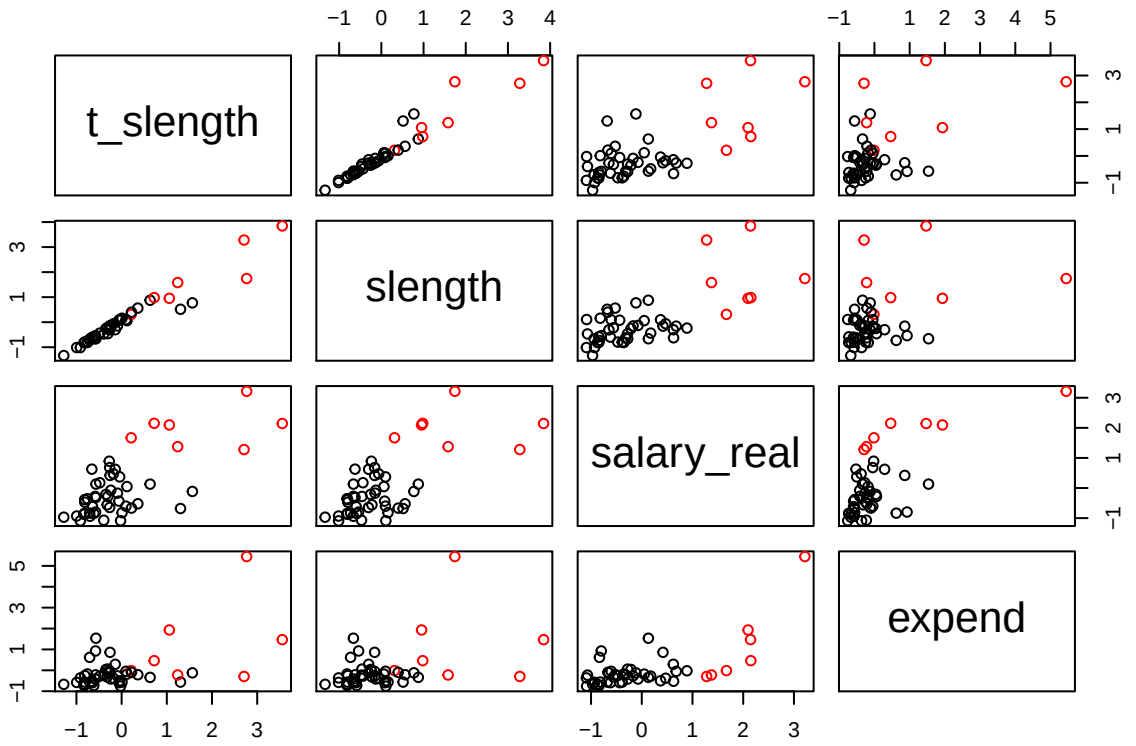
```
mcl <- Mclust(x_clean, 2)
summary(mcl)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model with 2
## components:
##
##  log-likelihood  n df       BIC       ICL
##       -84.48984 48 29 -281.2445 -281.7949
##
## Clustering table:
##  1  2
## 38 10
```

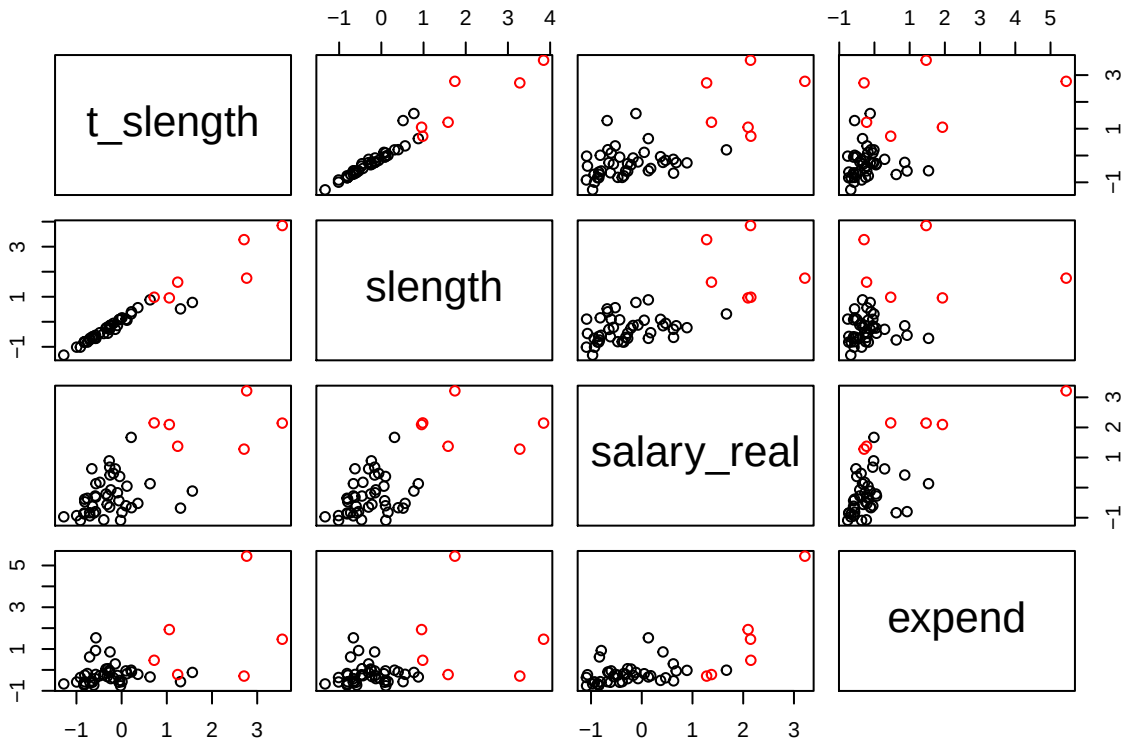
According to the clustering table, the mixture model was more generous with the smaller cluster, putting 10 observations in the smaller cluster, leaving 38 states in the larger cluster.

Question 7

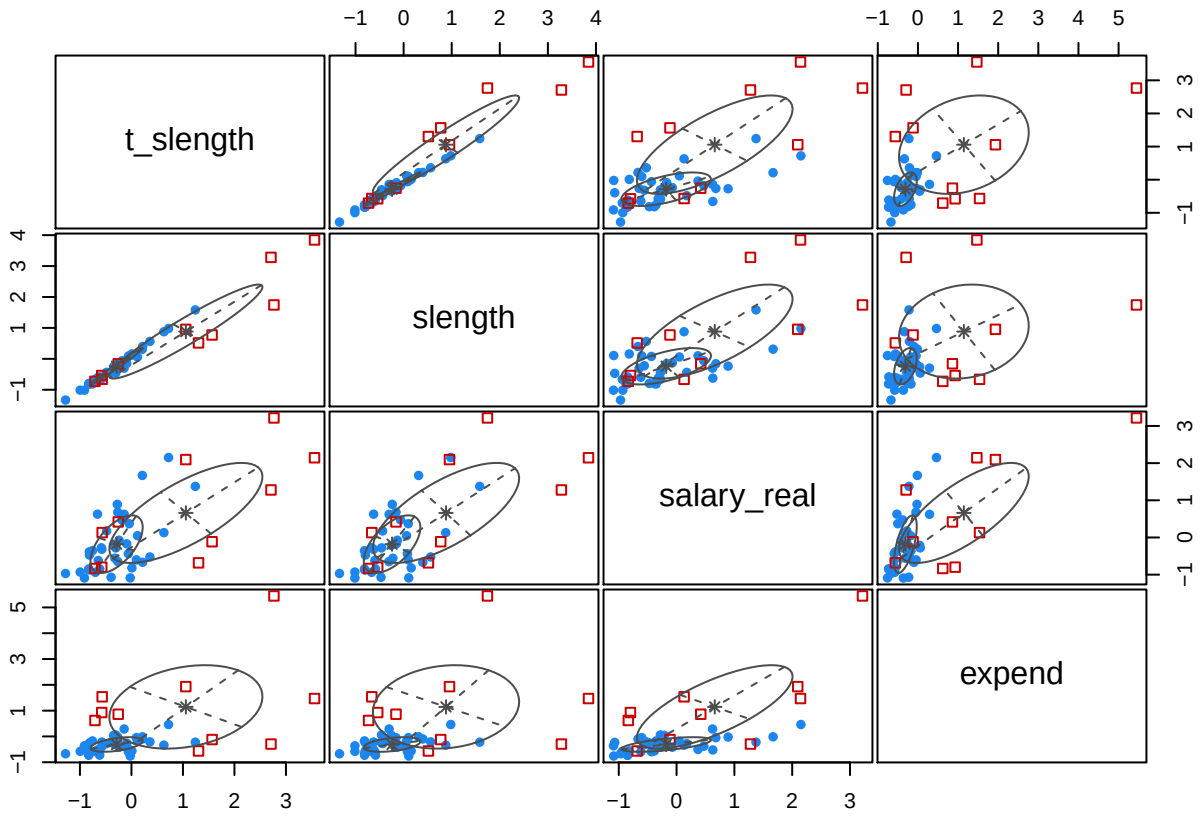
```
with(x_clean, pairs(x_clean, col=hc.cut$cluster))
```




```
with(x_clean, pairs(x_clean, col=c(1:4)[km$cluster]))
```



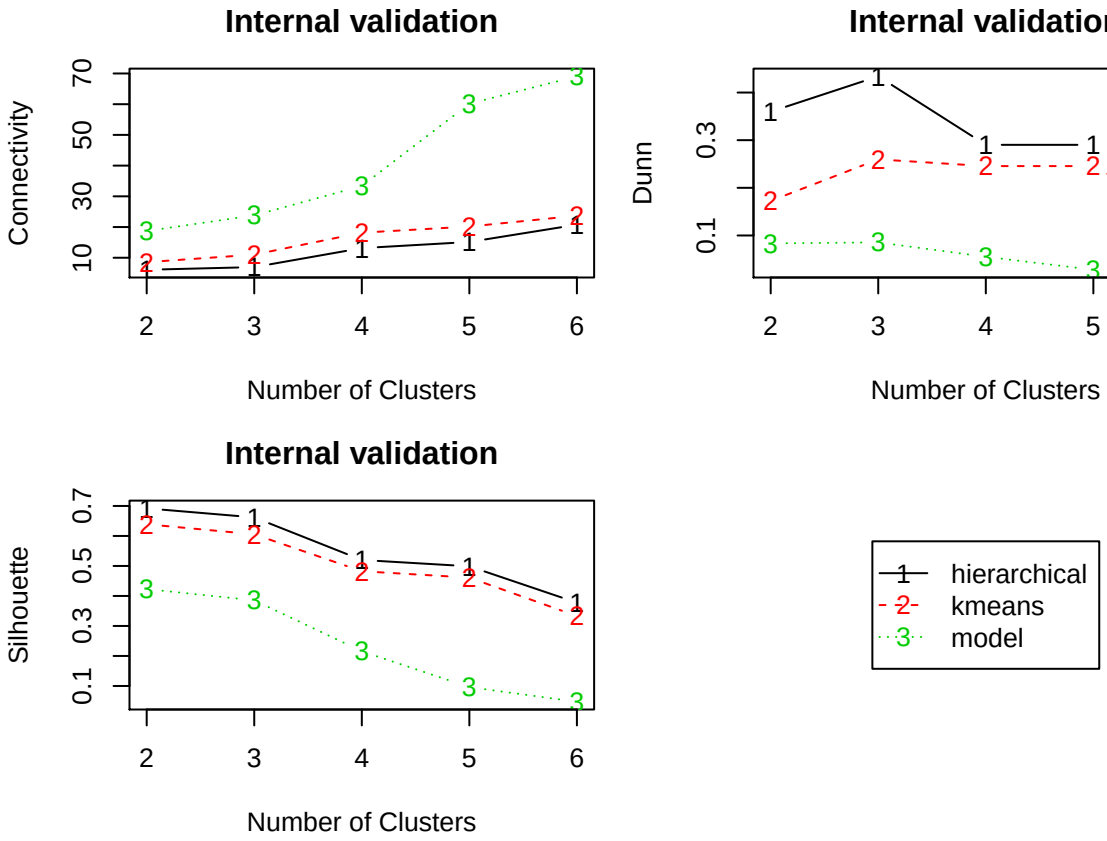

```
plot(mcl, what = "classification")
```



Question 8

```
cl_val <- clValid(x_clean, 2:6, clMethods=c("hierarchical","kmeans","model"),
        validation="internal")
```

```
op <- par(no.readonly = TRUE)
par(mfrow = c(2, 2), mar = c(4, 4, 3, 1))
plot(cl_val, legend = FALSE)
plot(nClusters(cl_val), measures(cl_val, "Dunn")[, , 1], type = "n",
     axes = F, xlab = "", ylab = "")
legend("center", clusterMethods(cl_val), col = 1:3, lty = 1:3,
       pch = paste(1:3))
```




```
par(op)
```

Question 9

```
summary(cl_val)
```

```
##
## Clustering Methods:
##  hierarchical kmeans model
##
## Cluster sizes:
##  2 3 4 5 6
##
## Validation Measures:
##                                  2       3       4       5       6
##
## hierarchical Connectivity   6.0869  6.9536 13.1345 15.1345 20.7563
##              Dunn           0.3598  0.4340  0.2902  0.2902  0.2836
##              Silhouette     0.6920  0.6619  0.5199  0.4989  0.3776
## kmeans       Connectivity   8.5683 11.0183 18.1651 20.1651 23.6810
##              Dunn           0.1726  0.2597  0.2456  0.2456  0.1214
##              Silhouette     0.6390  0.6054  0.4824  0.4611  0.3328
## model        Connectivity  18.7095 23.7964 33.3683 60.1651 69.0651
##              Dunn           0.0833  0.0855  0.0554  0.0280  0.0391
##              Silhouette     0.4230  0.3854  0.2157  0.0962  0.0473
##
```

```
## Optimal Scores:
##
##              Score  Method       Clusters
## Connectivity 6.0869 hierarchical 2
## Dunn         0.4340 hierarchical 3
## Silhouette   0.6920 hierarchical 2
```

```
optimalScores(cl_val)
```

```
##                  Score       Method Clusters
## Connectivity 6.0869048 hierarchical        2
## Dunn         0.4339669 hierarchical        3
## Silhouette   0.6920057 hierarchical        2
```

Overall, across the methods, 2 clusters seems to be the optimal number of clusters. In terms of which method seems to be preforming the best, it looks like the agglomerative hierarchical clustering algorithm (hc) seems to be getting the best internal validation scores. Looking at the results from the `optimalScores` function, we can see that the hc preformed best for all the internal validation methods. I am, though, interested in understanding more what is going on with the Dunn Index score for the hc algorithm, since the best score for this validation method is hc with 3 clusters, rather than 2. This is an example of when we might use a 'sub-optimal' clustering method (i.e. cutting at 2, when the Dunn index is showing us that it is optimal to be cut at 3) because we have strong priors indicating that the number of clusters should be 2 rather than 3, so we would use 2.