Assignment1

Shuyi Yu

10/27/2019

Question1

```
#Load the state legislative professionalism data from the folder
(load("/Users/Shuyi/Desktop/AS1code/legprof-components.v1.0.RData"))
```

```
## [1] "x"
```

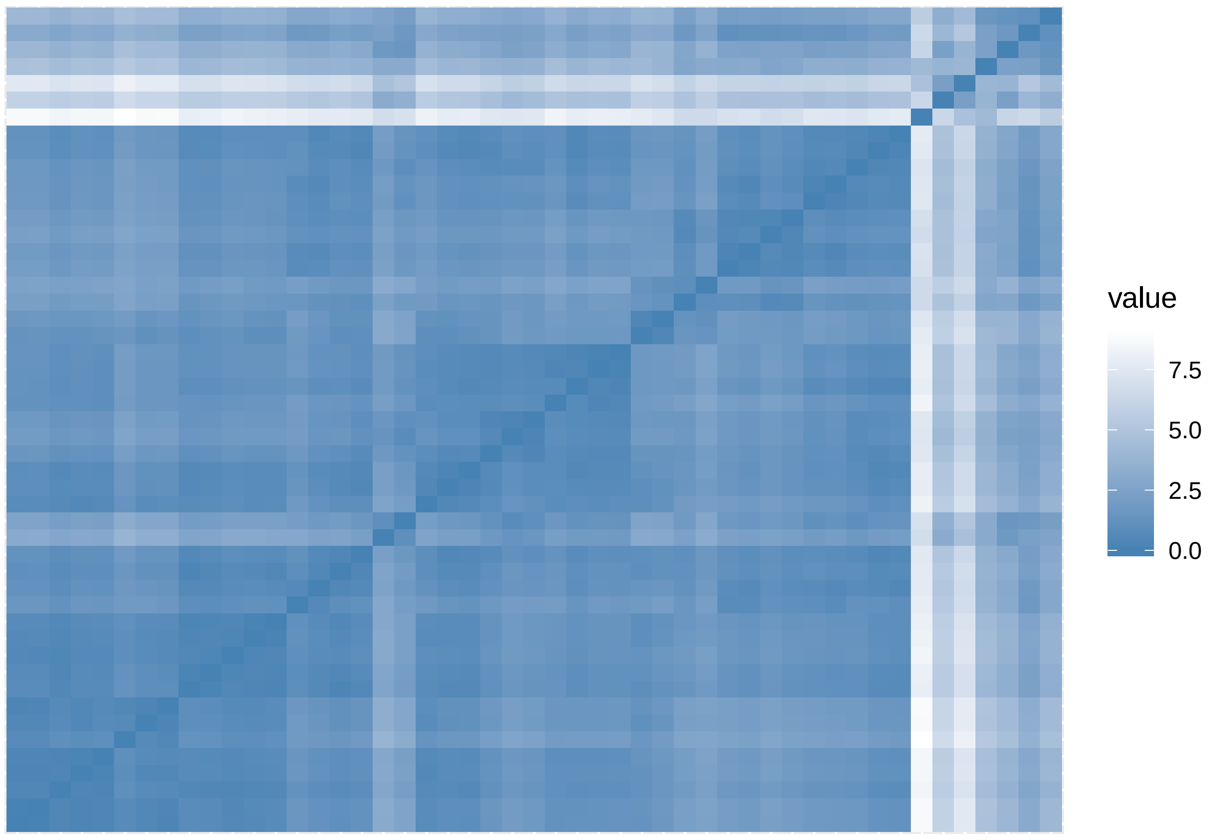
Question2

```
#Munge the data
d1 <- subset(x, sessid=="2009/10")
state_names <- d1[,3]
d2 <- d1 %>% select(t_slength, slength, salary_real, expend)
rownames(d2) <- state_names
d3 <- na.omit(d2)
pf_data <- d3 %>% scale()
```

Question3

From the graph, it's easy to see there is a natural, non-random structure in the data. There are darker blocks along the diagonal, which suggest greater spatial similarity, compared to lighter shaded blocks, which suggest greater dissimilarity.

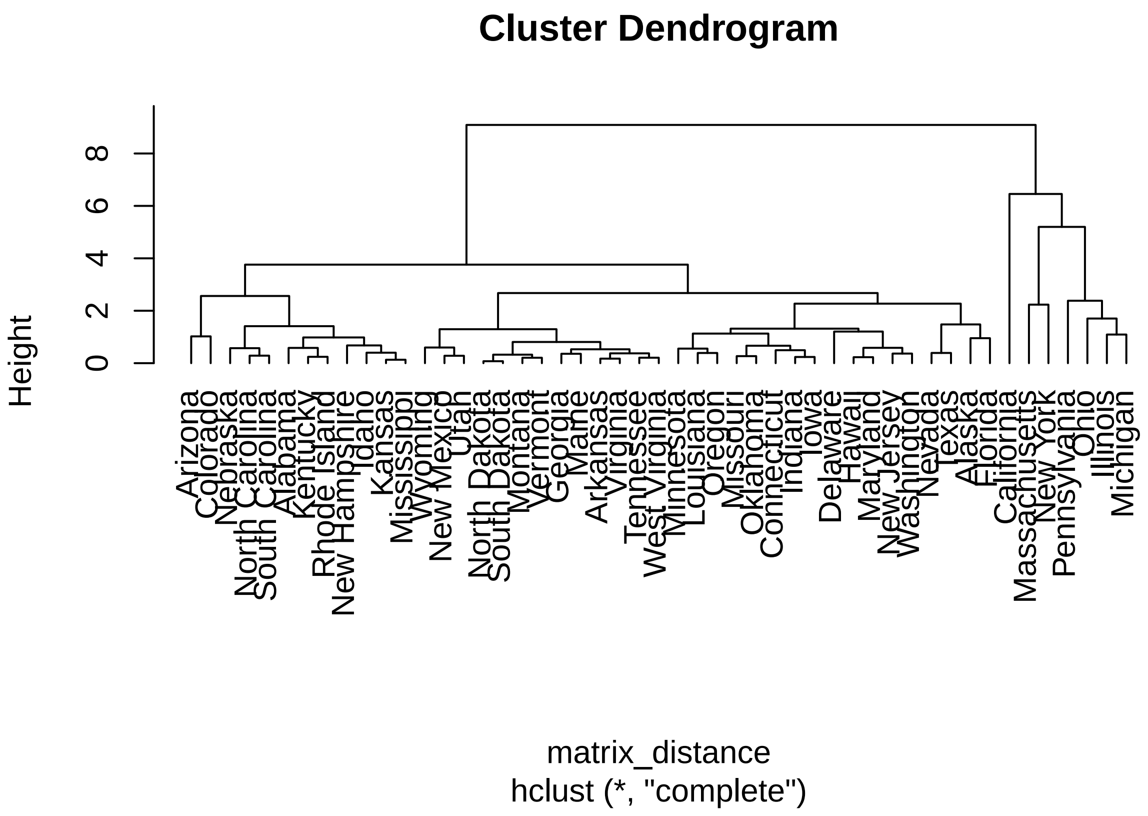
```
#Diagnose clusterability using ODI
clustend <- get_clust_tendency(pf_data, 48)
clustend$plot + scale_fill_gradient(low="steelblue", high="white")
```



Question4

The dendrogram can be first cut into two clusters, then further into three clusters. It fits certain observable natural grouping. I can see that the rightmost cluster includes populous wealthy states like California, Massachusetts, New York, Pennsyviania, Ohio, Illinois, Michigan, etc.

```r
#Fit a simple agglomerative hierarchical clustering algorithm
matrix_distance <- pf_data %>% dist()
pf_complete <- hclust(matrix_distance, method = "complete"); plot(pf_complete, hang = -1)
```



Question5

Let the Kmeans algorithm divide data into two clusters. The first cluster has a size of 43, with average t_slength -0.2930275 slength -0.2932285 salary -0.2833616 expend -0.2047966, representing states with shorter sessions and smaller legislative expenditure. The second cluster has a size of 6, with average t_slength 2.1000302 slength 2.1014710 salary 2.0307585 expend 1.4677087, representing states with longer sessions and larger legislative expenditure. From the cluster assignment, I can see that the second cluster has indeed populous wealthy states.

```r
#Fit a k-means algorithm
set.seed(634)
pf_kmeans <- kmeans(pf_data[,1:4], centers = 2, nstart = 15)
pf_kmeans$cluster
```

```
##        Alabama         Alaska        Arizona       Arkansas     California
##              1              1              1              1              2
##       Colorado    Connecticut       Delaware        Florida        Georgia
##              1              1              1              1              1
##         Hawaii          Idaho       Illinois        Indiana           Iowa
##              1              1              1              1              1
##         Kansas       Kentucky      Louisiana          Maine       Maryland
##              1              1              1              1              1
##  Massachusetts       Michigan      Minnesota    Mississippi       Missouri
##              2              2              1              1              1
##        Montana       Nebraska         Nevada  New Hampshire     New Jersey
##              1              1              1              1              1
##     New Mexico       New York North Carolina   North Dakota           Ohio
##              1              2              1              1              2
##       Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina
##               1              1              2              1              1
##    South Dakota      Tennessee          Texas           Utah        Vermont
##               1              1              1              1              1
##        Virginia     Washington  West Virginia        Wyoming
##               1              1              1              1
```

```
pf_kmeans$centers
```

```
##    t_slength    slength salary_real     expend
## 1 -0.2930275 -0.2932285  -0.2833616 -0.2047966
## 2  2.1000302  2.1014710   2.0307585  1.4677087
```

```
pf_kmeans$size
```

```
## [1] 43  6
```

Question6

Via μ, I can see that the first population has longer sessions and larger legislative expenditure, and the second population has shorter sessions and smaller legislative expenditure. Via λ, I can see that the second population contributes a larger proportion to the whole distribution.

```
##Fit a Gaussian mixture model via the EM algorithm
set.seed(7355)
pf_mgmm <- mvnormalmixEM(pf_data[,1:4], k = 2)
```

```
## number of iterations= 18
```

```
pf_mgmm[2:3]
```

```
## $lambda
## [1] 0.122743 0.877257
##
## $mu
## $mu[[1]]
## [1] 2.094531 2.095874 2.029265 1.463510
##
## $mu[[2]]
## [1] -0.2930600 -0.2932478 -0.2839281 -0.2047695
```

Question7

The dendrogram is drawn in Qustion4. I fit Gaussian mixture model on several combinations of two features. Here are two graphs of them. The left graph is on salary and expenditure, and the two distributions are not clear. I guess it's bacause salary and expenditure are too much overlapping, the two features cannot be used to cluster well. The right graph is on session length and salary, and the two distributions are very clear. Actually using session length and salaray produces the best fitting results across all combinations.

```
#Plot by state cluster assignment across two features like salary and expenditures
par(mfrow = c(1, 2))
set.seed(7355)
pf_mgmm <- mvnormalmixEM(pf_data[,3:4], k = 2)
```

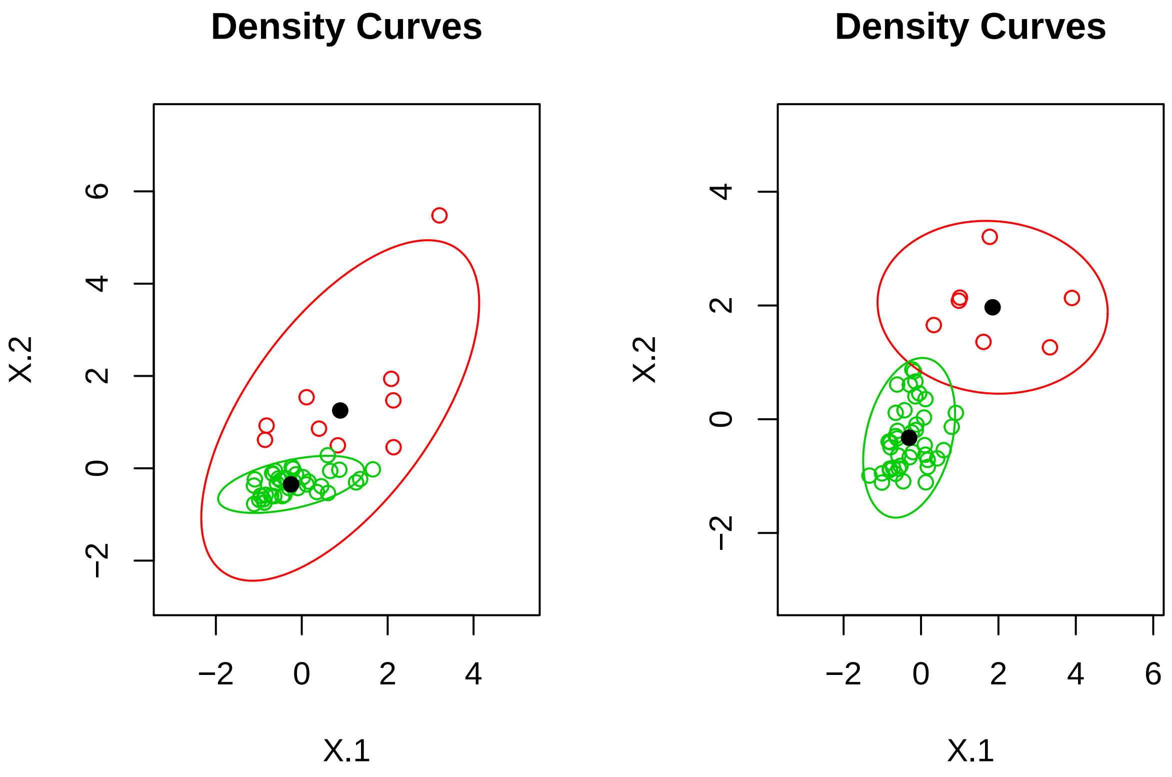
```
## number of iterations= 44
```

```
plot(pf_mgmm, which=2,
     xlim = c(min(pf_mgmm$x[,1])-2, max(pf_mgmm$x[,1])+2),
     ylim = c(min(pf_mgmm$x[,2])-2, max(pf_mgmm$x[,2])+2)
)

set.seed(7355)
pf_mgmm <- mvnormalmixEM(pf_data[,2:3], k = 2)
```

```
## number of iterations= 19
```

```
plot(pf_mgmm, which=2,
     xlim = c(min(pf_mgmm$x[,1])-2, max(pf_mgmm$x[,1])+2),
     ylim = c(min(pf_mgmm$x[,2])-2, max(pf_mgmm$x[,2])+2)
)
```



Question8

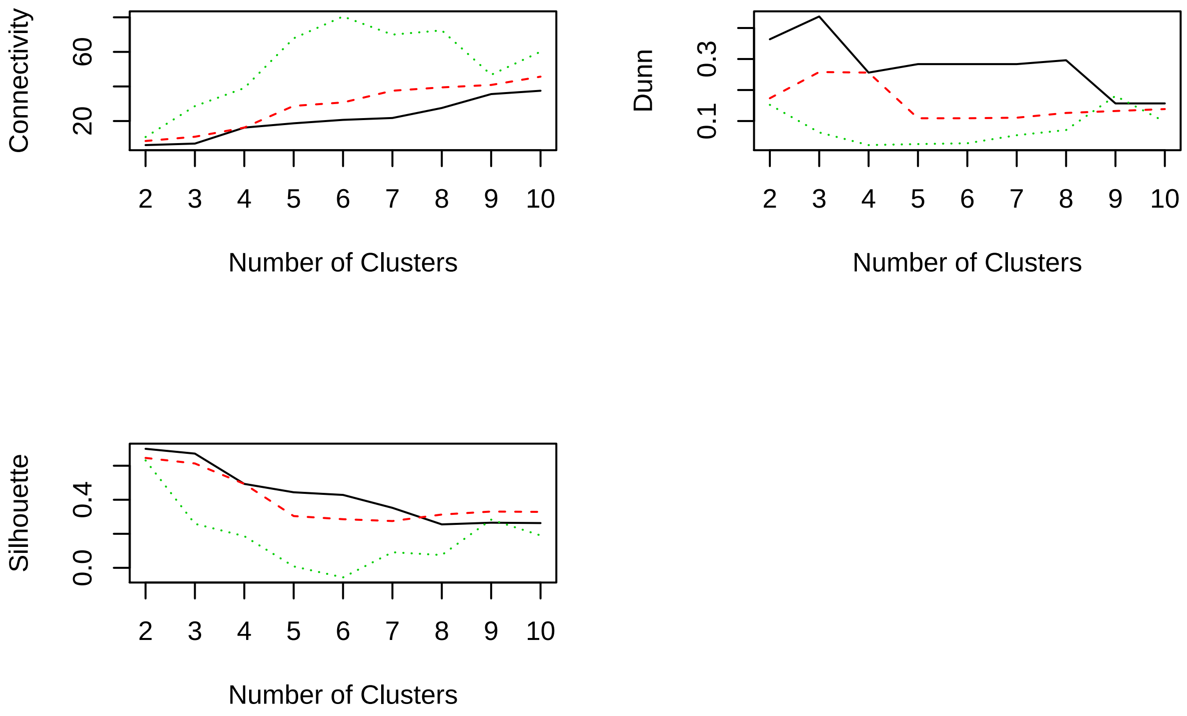
Let's look at Silhouette, which shows that the hierarchical algorithm is the best with k=2.

```
#Validation
pf_matrix <- as.matrix(pf_data[,1:4])
internal_all <-
  clValid(pf_matrix, 2:10, clMethods=c("hierarchical", "kmeans", "model"), validation="internal"); summa
```

```
##
## Clustering Methods:
##  hierarchical kmeans model
##
## Cluster sizes:
##  2 3 4 5 6 7 8 9 10
##
## Validation Measures:
##                                  2       3       4       5       6       7       8       9      10
##
```

```
## hierarchical Connectivity   6.0869  6.9536 16.1885 18.6774 20.6774 21.7607 27.5476 35.5813 37.5147
##              Dunn           0.3637  0.4371  0.2562  0.2836  0.2836  0.2836  0.2960  0.1568  0.1568
##              Silhouette     0.6994  0.6711  0.4932  0.4440  0.4284  0.3525  0.2553  0.2652  0.2630
## kmeans       Connectivity   8.4460 10.8960 16.1885 28.7437 30.7437 37.5266 39.4552 40.8694 45.6623
##              Dunn           0.1735  0.2581  0.2562  0.1090  0.1090  0.1108  0.1260  0.1324  0.1386
##              Silhouette     0.6458  0.6131  0.4932  0.3042  0.2858  0.2750  0.3131  0.3307  0.3288
## model        Connectivity  10.7393 28.6119 39.0687 67.8401 80.4806 69.9774 72.4377 46.7254 60.0976
##              Dunn           0.1522  0.0633  0.0225  0.0258  0.0283  0.0543  0.0710  0.1810  0.0977
##              Silhouette     0.6314  0.2588  0.1861  0.0085 -0.0562  0.0917  0.0752  0.2831  0.1905
##
## Optimal Scores:
##
##              Score  Method       Clusters
## Connectivity 6.0869 hierarchical 2
## Dunn         0.4371 hierarchical 3
## Silhouette   0.6994 hierarchical 2
```

```r
par(mfrow = c(2, 2))
plot(internal_all, legend = FALSE, type = "l", main = " ")
```



Question9

a. For this context, in terms of better validation statistics, hierarchical > kmeans > model.
b. Hierarchical algorithm is the best with k=2.
c. First, hierarchical algorithm is computationally intensive and does not scale well, so if there is huge dataset, we may need to employ less demanding method like kmeans. Second, sometimes, we do not have domain knowledge to infer a good k, thus hierarchical algorithm will be better in exploring the underlying structure of the dataset.