ps-4

Shuai Yuan

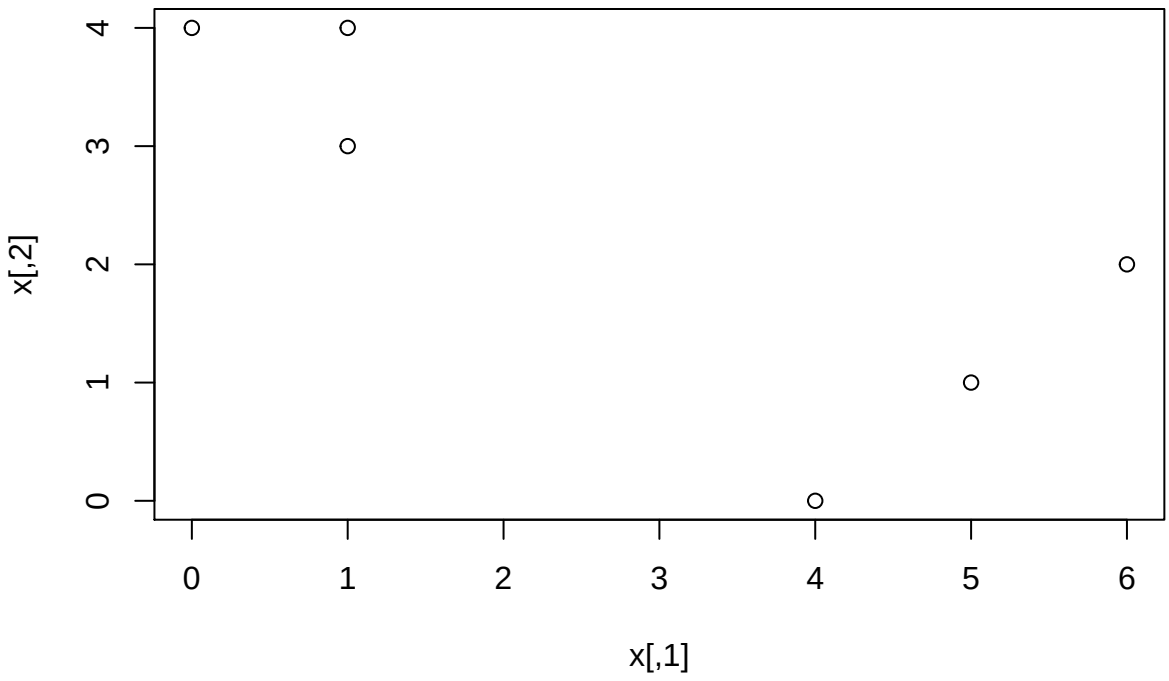
3/2/2020

Q1

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

1.1

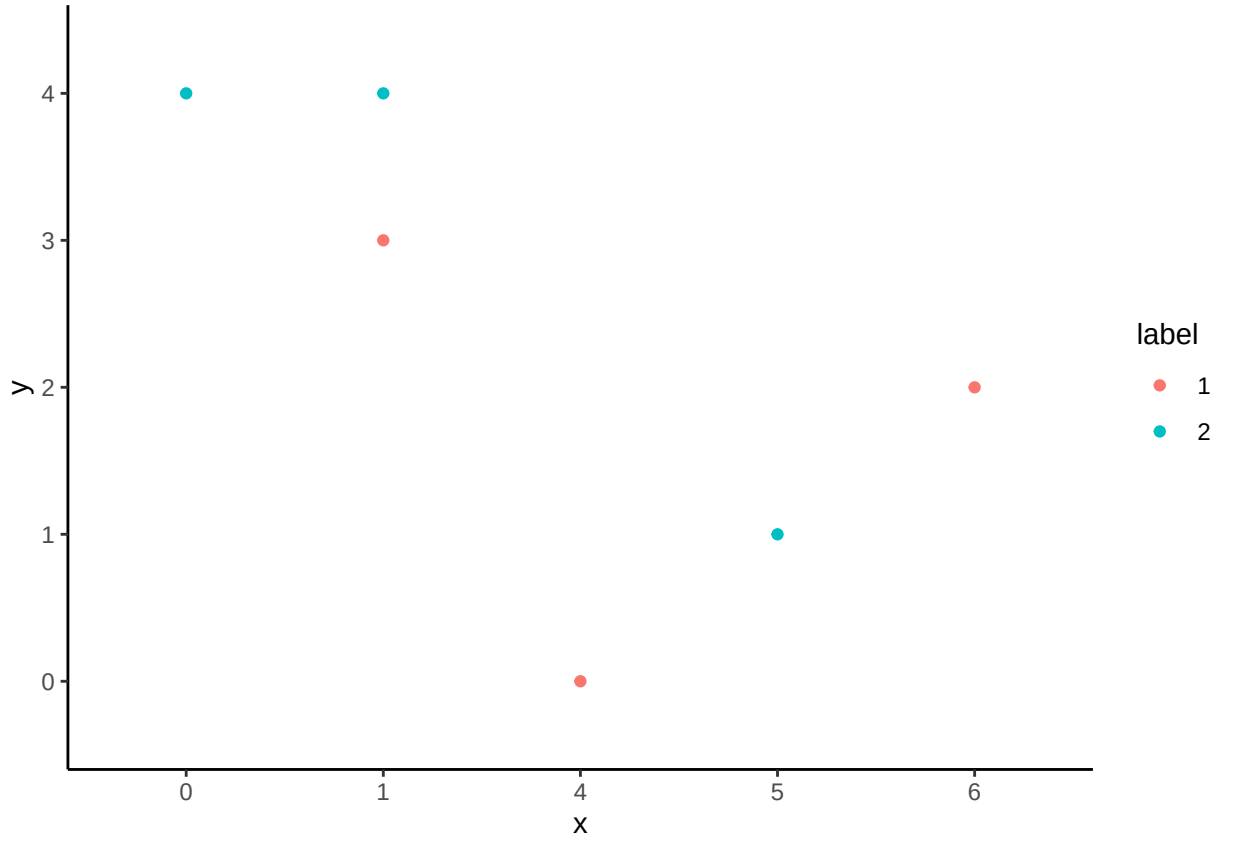
```
plot(x)
```



1.2

```
set.seed(1345)
label <- sample(c('1','2'), size = nrow(x),replace = TRUE)
cluster <- cbind(x, label)%>%
  data.frame()%>%
  rename(x = V1,
         y = V2)
```

```
cluster%>%
  ggplot(aes(x, y, color = label)) +
  geom_point()
```



1.3

```
cluster<- cluster%>%
  mutate(x = as.numeric(x))%>%
  mutate(y = as.numeric(y))%>%
  mutate(label = as.numeric(label))

cluster%>%
  group_by(label)%>%
  summarize(x = mean(x),
            y = mean(y))%>%
  kable()
```

label	x	y
1	3.333333	2.666667
2	2.333333	4.000000

1.4

```
assign<- function(data){
  centroid<- data%>%
    group_by(label)%>%
    summarize(x = mean(x),
```

```
              y = mean(y))
  data <- data%>% mutate(label = ifelse(
    ((data$x - centroid$x[1])^2 + (data$y - centroid$y[1])^2)<
    ((data$x - centroid$x[2])^2 + (data$y - centroid$y[2])^2),
    1,2))
  return(data)
}

new<- assign(cluster)
kable(new)
```

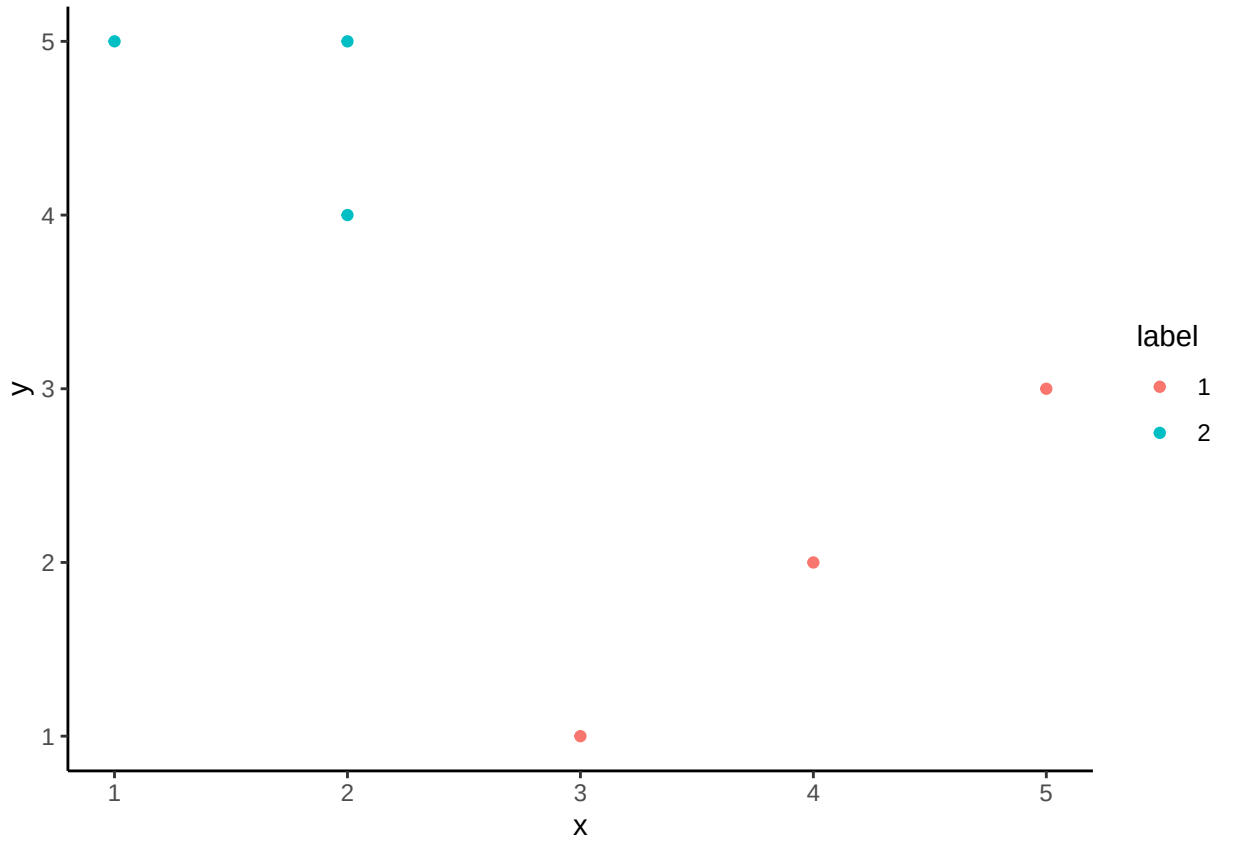
x	y	label
2	5	2
2	4	2
1	5	2
4	2	1
5	3	1
3	1	1

1.5

```
for (i in 1:1000){
old_label <- cluster$label

centroid<- cluster%>%
    group_by(label)%>%
    summarize(x = mean(x),
              y = mean(y))

for (i in 1:nrow(cluster)){
  dis.1<- (cluster$x[i] - centroid$x[1])^2 + (cluster$y[i] - centroid$y[1])^2
  dis.2<- (cluster$x[i] - centroid$x[2])^2 + (cluster$y[i] - centroid$y[2])^2
  if (dis.1 < dis.2) {cluster$label[i]<- 1}
  else{cluster$label[i]<- 2}
}
new_label<- cluster$label

if(sum(old_label)-sum(new_label)== 0){break}}
```

1.6

```
cluster%>%
  mutate(label = as.factor(label))%>%
  ggplot(aes(x,y,color=label))+
  geom_point()
```



Question 2

2.1

```
load("./Data and Codebook/legprof-components.v1.0.RData")
```

2.2

```
df <- x%>%
  as.data.frame()%>%
  select(stateabv, sessid, t_slength, slength,salary_real,expend)%>%
  filter(sessid == "2009/10")%>%
  na.omit()

df[,3:6]<- scale(df[,3:6])

row.names(df)<- df$stateabv
df<- df[,3:6]
```

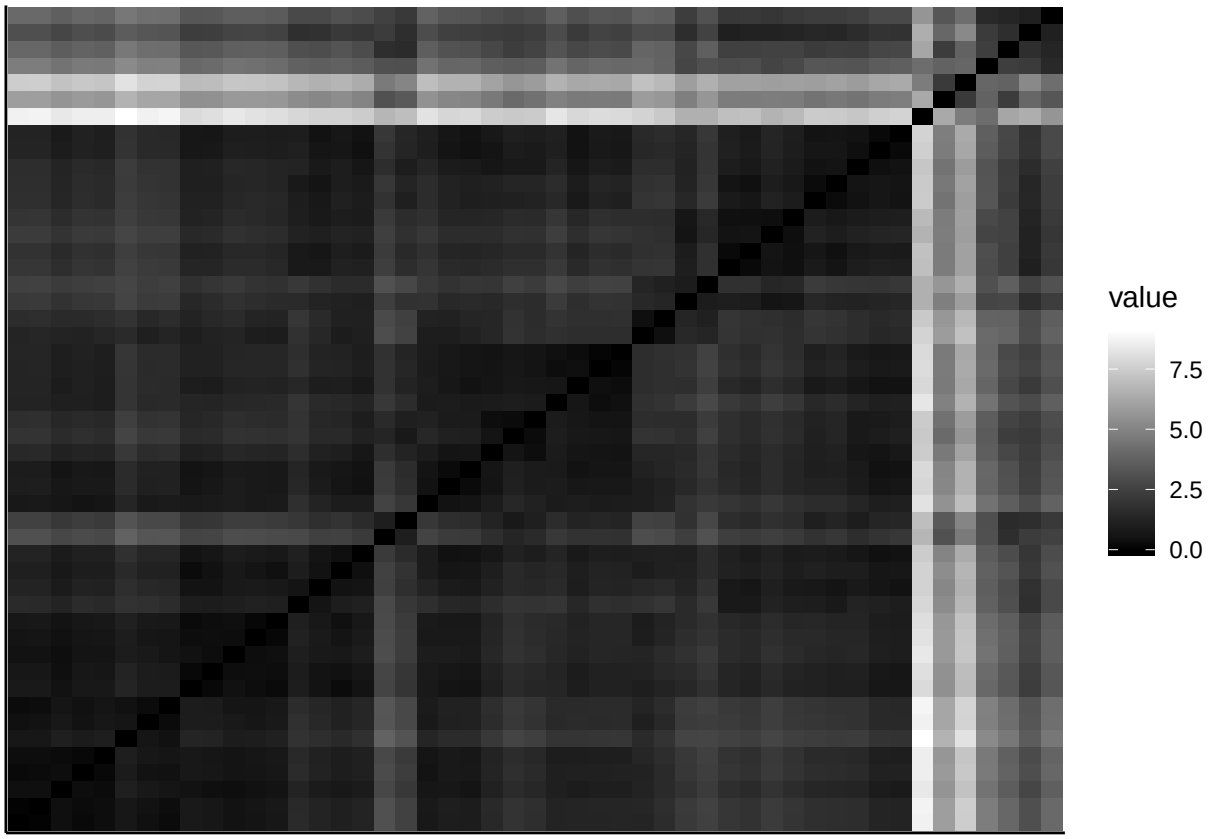
2.3

By looking at the graph, we can somewhat see a diagonal, but we still can't decide if the data is non-random and clusterable by simply looking at the graph. But if we look at the Hopkins Statistics of 0.841, we know that the data is highly clustered.

```
gradient_list = list(low = "black", high = "white")
get_clust_tendency(df, n=40, gradient = gradient_list)
```

```
## $hopkins_stat
```

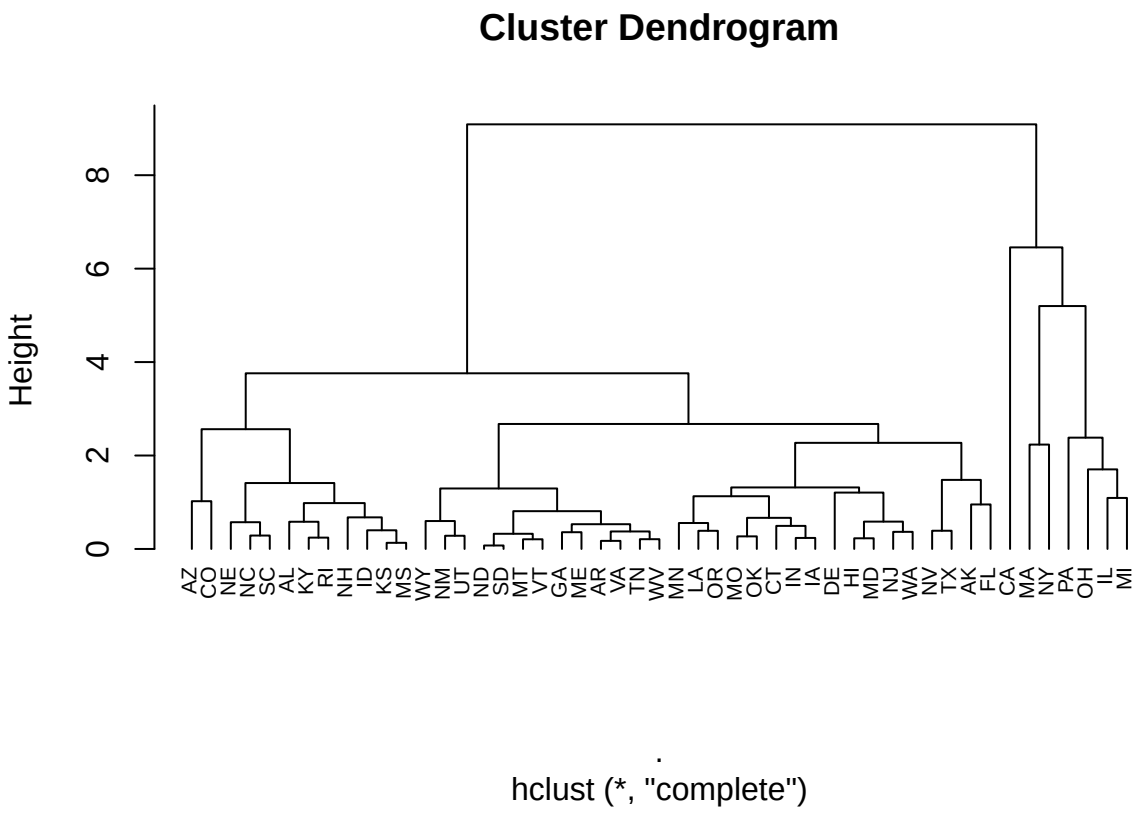
```
## [1] 0.8406165
##
## $plot
```



2.4

The dendrogram shows that two large clusters are identidied. Within each large cluster, smaller clusters are identified based on geographical positions.

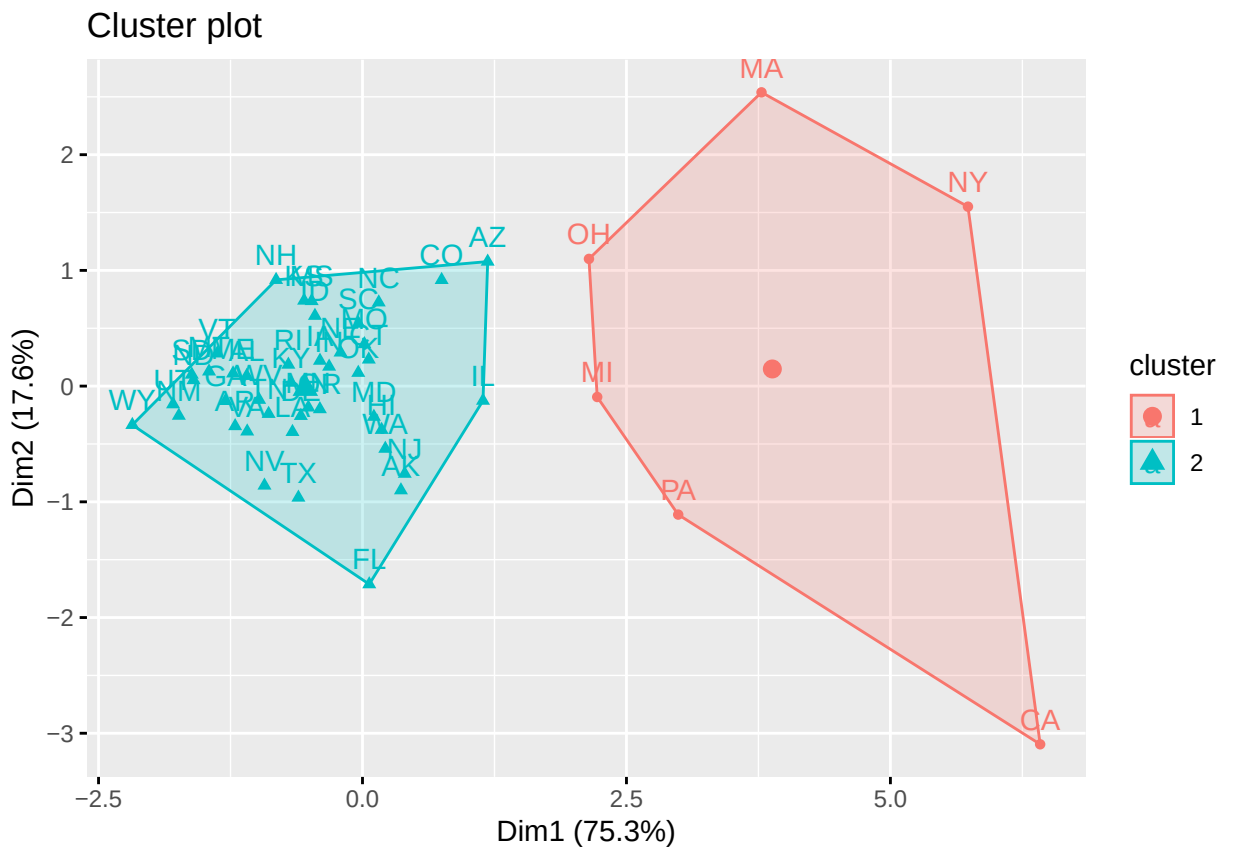
```
df_tree<- df%>%
  dist()%>%
  hclust(method = "complete")
plot(df_tree, cex=0.7, hang=-1)
```



2.5

Though a different clustering strategy is used here, we still get similar cluster results. Only Illinois is being assigned to the other cluster.

```
k2<- kmeans(df, centers =2, nstart =15)
fviz<- fviz_cluster(k2, df)
fviz
```



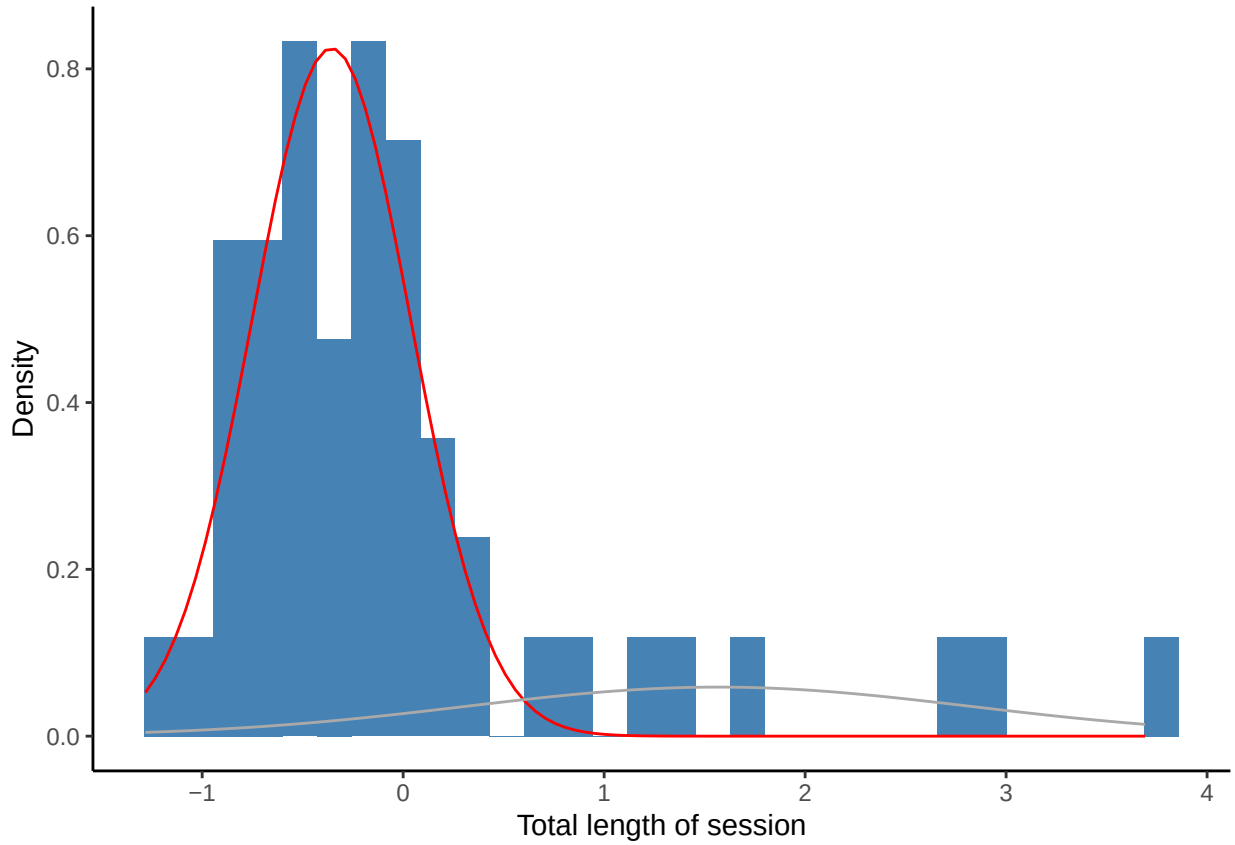
2.6

By looking at the plot, we can see a similar cluster classification as well. Still the majority of the states are clustered under the red curve. However, there are more states identified in the relatively smaller cluster.

```
gmm1<- normalmixEM(df$t_slength, k=2)
```

```
## number of iterations= 26
```

```
p1<- ggplot(data.frame(x = gmm1$x)) +
  geom_histogram(aes(x,..density..),fill = "steelblue")+
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm1$mu[1], gmm1$sigma[1],lam = gmm1$lambda[1]),
                color = "red")+
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm1$mu[2], gmm1$sigma[2],lam = gmm1$lambda[2]),
                color = "darkgray")+
  xlab("Total length of session")+
  ylab("Density")
p1
```



2.7

By looking at the results from GMM methods, there is not much difference in the cluster classification except for the real wage one, which is significantly different from the others.

```
set.seed(123)
gmm2<- normalmixEM(df$slength, k=2)
```

```
## number of iterations= 49
```

```
p2<- ggplot(data.frame(x = gmm2$x)) +
  geom_histogram(aes(x,..density..),fill = "steelblue")+
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm2$mu[1], gmm2$sigma[1],lam = gmm2$lambda[1]),
                color = "red")+
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm2$mu[2], gmm2$sigma[2],lam = gmm2$lambda[2]),
                color = "darkgray")+
  xlab("Length of Regular Session")+
  ylab("Density")

gmm3<- normalmixEM(df$salary_real, k=2)
```

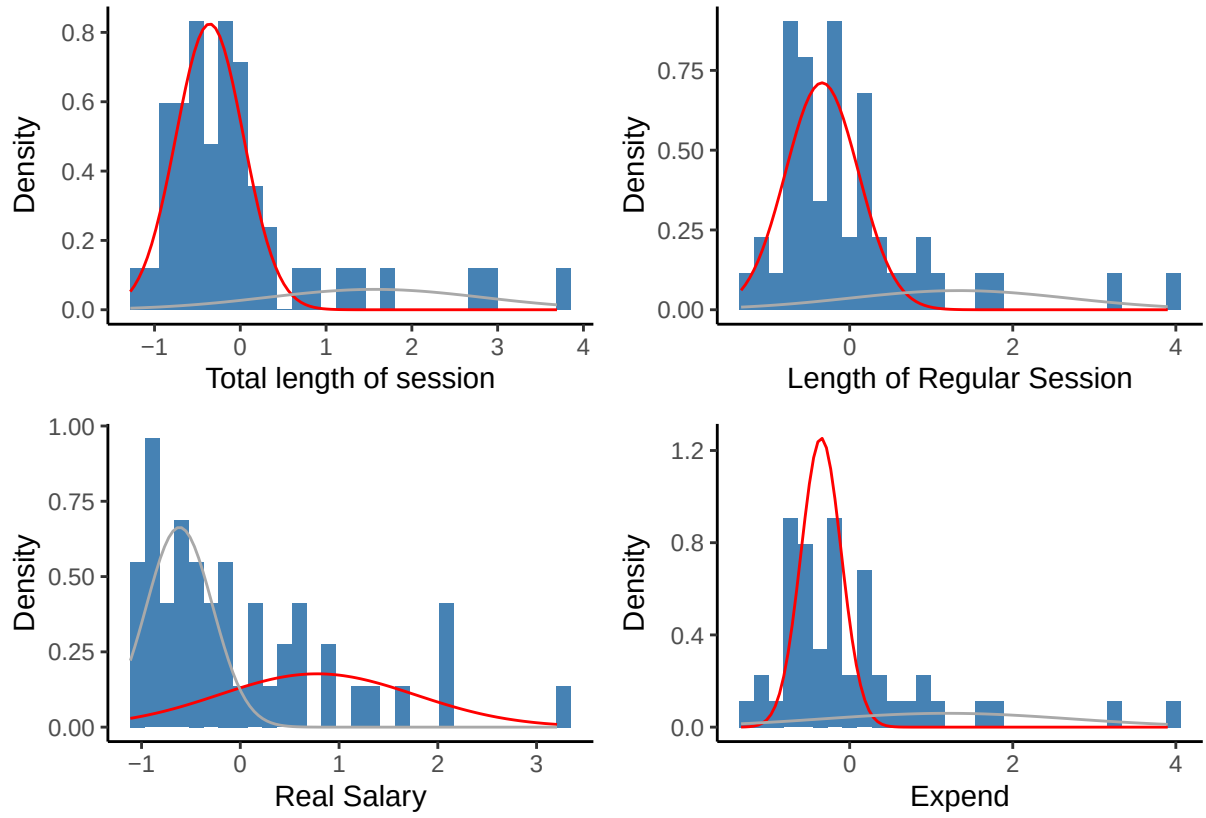
```
## number of iterations= 45
```

```
p3<- ggplot(data.frame(x = gmm3$x)) +
  geom_histogram(aes(x,..density..),fill = "steelblue")+
  stat_function(geom = "line", fun = plot_mix_comps,
```

```
                args = list(gmm3$mu[1], gmm3$sigma[1],lam = gmm3$lambda[1]),
                color = "red")+
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm3$mu[2], gmm3$sigma[2],lam = gmm3$lambda[2]),
                color = "darkgray")+
  xlab("Real Salary")+
  ylab("Density")

gmm4<- normalmixEM(df$expend, k=2)
```

```
## number of iterations= 14
```

```
p4<- ggplot(data.frame(x = gmm2$x)) +
  geom_histogram(aes(x,..density..),fill = "steelblue")+
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm4$mu[1], gmm4$sigma[1],lam = gmm4$lambda[1]),
                color = "red")+
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm4$mu[2], gmm4$sigma[2],lam = gmm4$lambda[2]),
                color = "darkgray")+
  xlab("Expend")+
  ylab("Density")

p1+p2+p3+p4
```



2.8

By comparing across different clmethods, the optimal scores and methods are listed as follows.

Connectivity Index corresponds to what extent items are placed in the same cluster as their nearest neighbors in the data space. The connectivity has a value between 0 and infinity and should be minimized.

Silhouette Index shows if the object matches its own cluster and if the object matches its neighboring cluster or not. The range is between -1 and 1. The higher the value, the better the object fits its own cluster and worse it fits neighboring cluster.

Dunn Index is actually the ratio between the minimum of this pairwise distance as the inter-cluster separation (min.separation) and the maximal intra-cluster distance (i.e maximum diameter) as the intra-cluster compactness. It should be as small as possible.

By looking at the score, the connectivity score is a bit high for being over 7, the silhouette score is also a bit high for being over 0.6. Though the Dunn score is not as small as we expected, I personally think it is still the best choice among the three.

```
valid<- clValid(df, 2:15, clMethods = c("hierarchical","kmeans","model"),
                validation= "internal", method = "complete")
summary(valid)
```

```
##
## Clustering Methods:
##  hierarchical kmeans model
##
## Cluster sizes:
##  2 3 4 5 6 7 8 9 10 11 12 13 14 15
##
## Validation Measures:
##                                  2       3       4       5       6       7       8       9      10
##
## hierarchical Connectivity   7.9071 10.5238 12.9583 25.7397 31.7056 34.9440 37.2218 42.7048 44.7048 46
##              Dunn           0.1673  0.2077  0.2872  0.1731  0.1094  0.1177  0.1235  0.1255  0.1647  0
##              Silhouette     0.6204  0.5884  0.5236  0.2391  0.3125  0.3057  0.2996  0.3362  0.3213  0
## kmeans       Connectivity   8.4460 10.8960 16.1885 28.7437 35.1774 38.4913 39.3079 43.6623 45.6623 47
##              Dunn           0.1735  0.2581  0.2562  0.1090  0.1130  0.1181  0.1206  0.1057  0.1386  0
##              Silhouette     0.6458  0.6131  0.4932  0.3042  0.3388  0.3267  0.3216  0.3437  0.3288  0
## model        Connectivity  10.7393 28.6119 39.0687 67.8401 80.4806 69.9774 72.4377 46.7254 60.0976 60
##              Dunn           0.1522  0.0633  0.0225  0.0258  0.0283  0.0543  0.0710  0.1810  0.0977  0
##              Silhouette     0.6314  0.2588  0.1861  0.0085 -0.0562  0.0917  0.0752  0.2831  0.1905  0
##
## Optimal Scores:
##
##              Score  Method       Clusters
## Connectivity 7.9071 hierarchical 2
## Dunn         0.2872 hierarchical 4
## Silhouette   0.6458 kmeans       2
```

2.9

We can see clearly that different internal validation method will tell us different answers to the question that which method of clustering should we be using. This is happening they have different ways of making compromise between similarity within cluster and dissimilarity across clusters. Therefore, it is generally hard to come to the conclusion which method is optimal.

There are three optimal approaches depending on different internal validation measure. Hierarchical method with either 2 clusters or 4 clusters are the best according to Connectivity and Dunn respectively. kmeans with 2 clusters is the best according to Silhouette.

It is highly likely that we choose a sub-optimal approach. It depends on the context of the research problem and the priority of the measure(either the similarity within cluster and dissimilarity across clusters).