Problem Set 4

Shuai Yuan

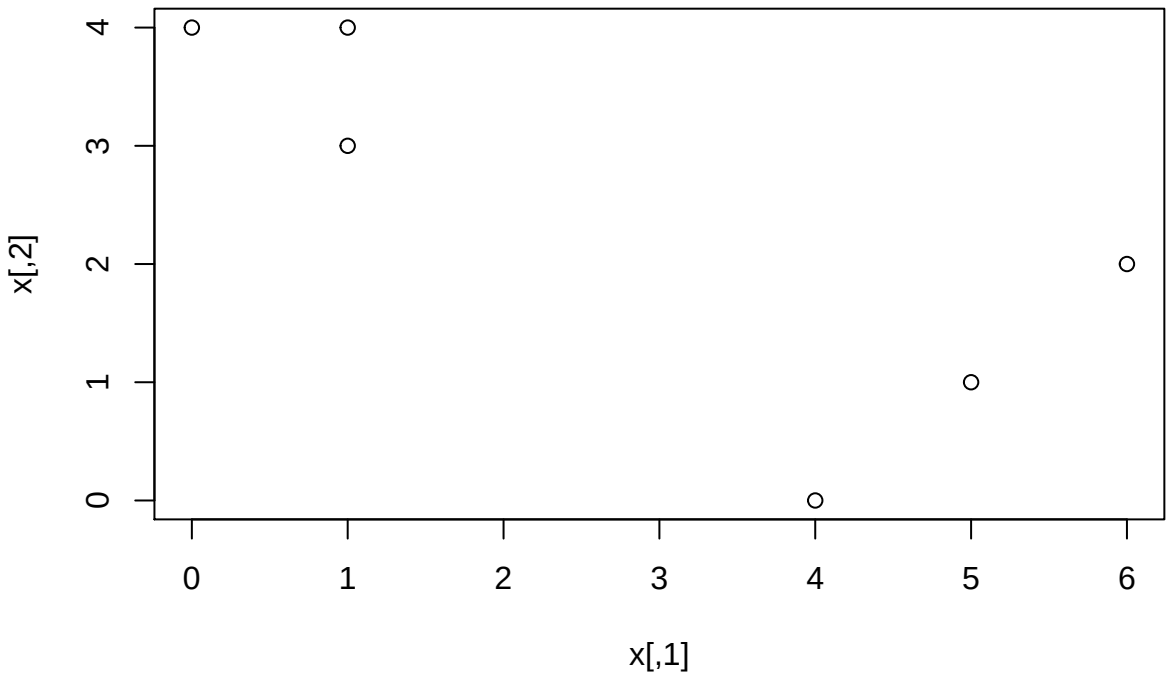
March 1, 2020

Q1

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

1.1

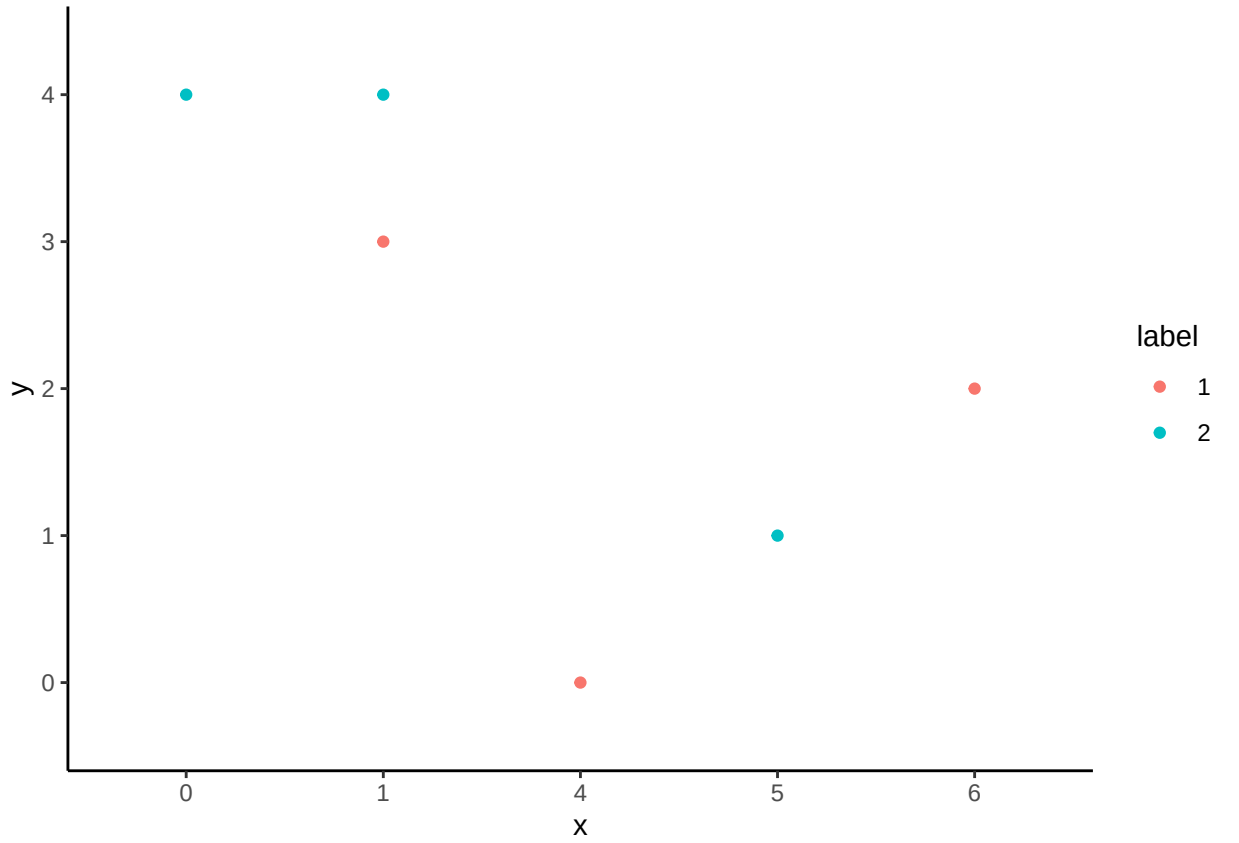
```
plot(x)
```



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)
```

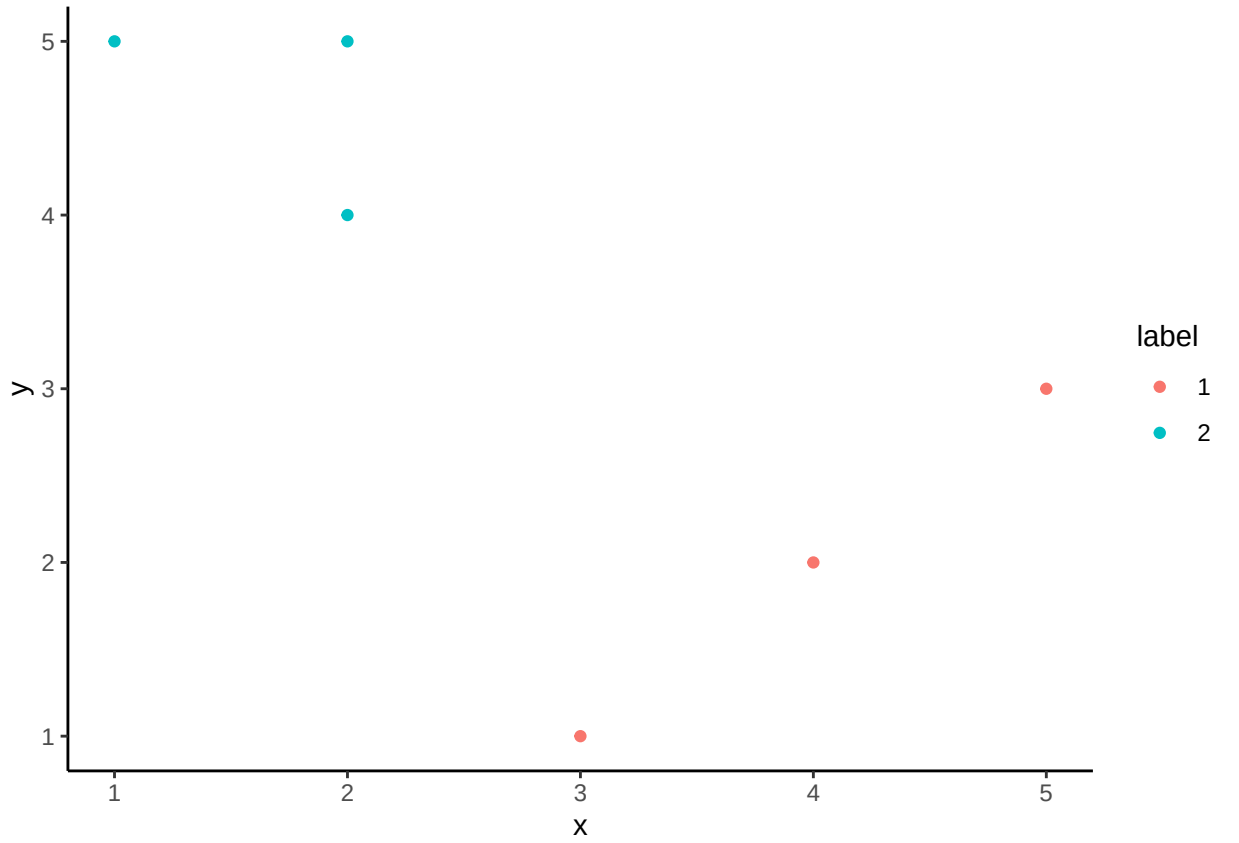
x	y	label
2	5	2
2	4	2
1	5	2
4	2	1
5	3	1
3	1	1

1.5

{r} same = F while(!same){ cluster<- mutate(cluster, label = as.character(label)) new<- mutate(new, label = as.character(label)) same<- all(cluster$label == new$label) }

1.6

```
new%>%
  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

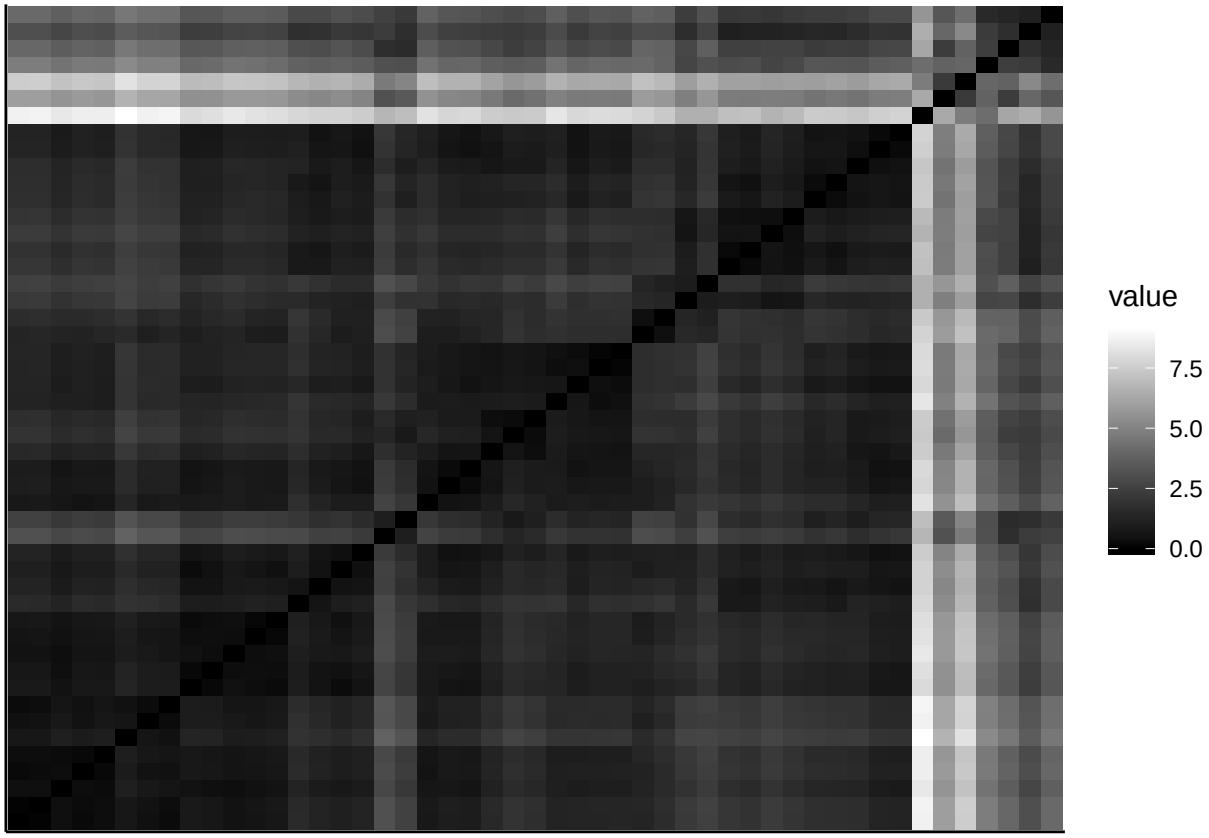
```
df <- x%>%
  as.data.frame()%>%
  select(stateabv, sessid, t_slength, slength,salary_real,expend)%>%
  filter(sessid == "2009/10")%>%
  na.omit()

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, we know that the data is highly clustered.

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

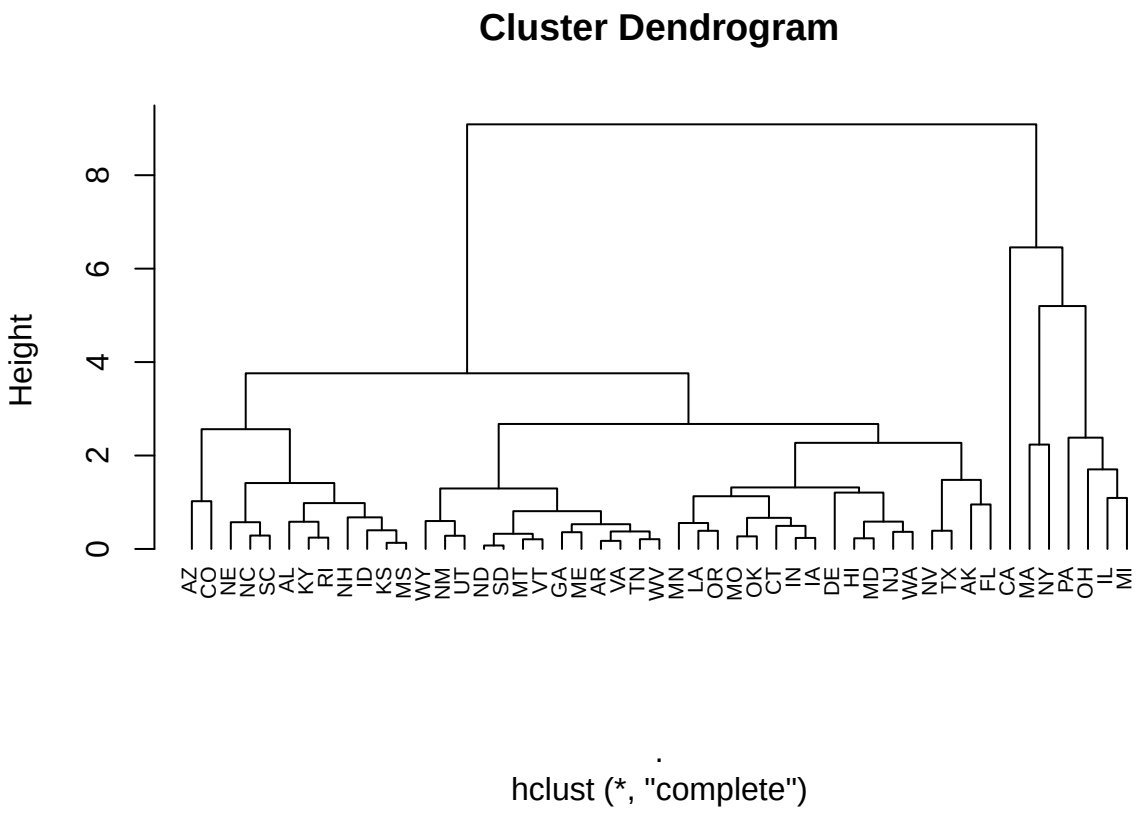
```
## $hopkins_stat
## [1] 0.8406165
##
## $plot
```



2.4

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

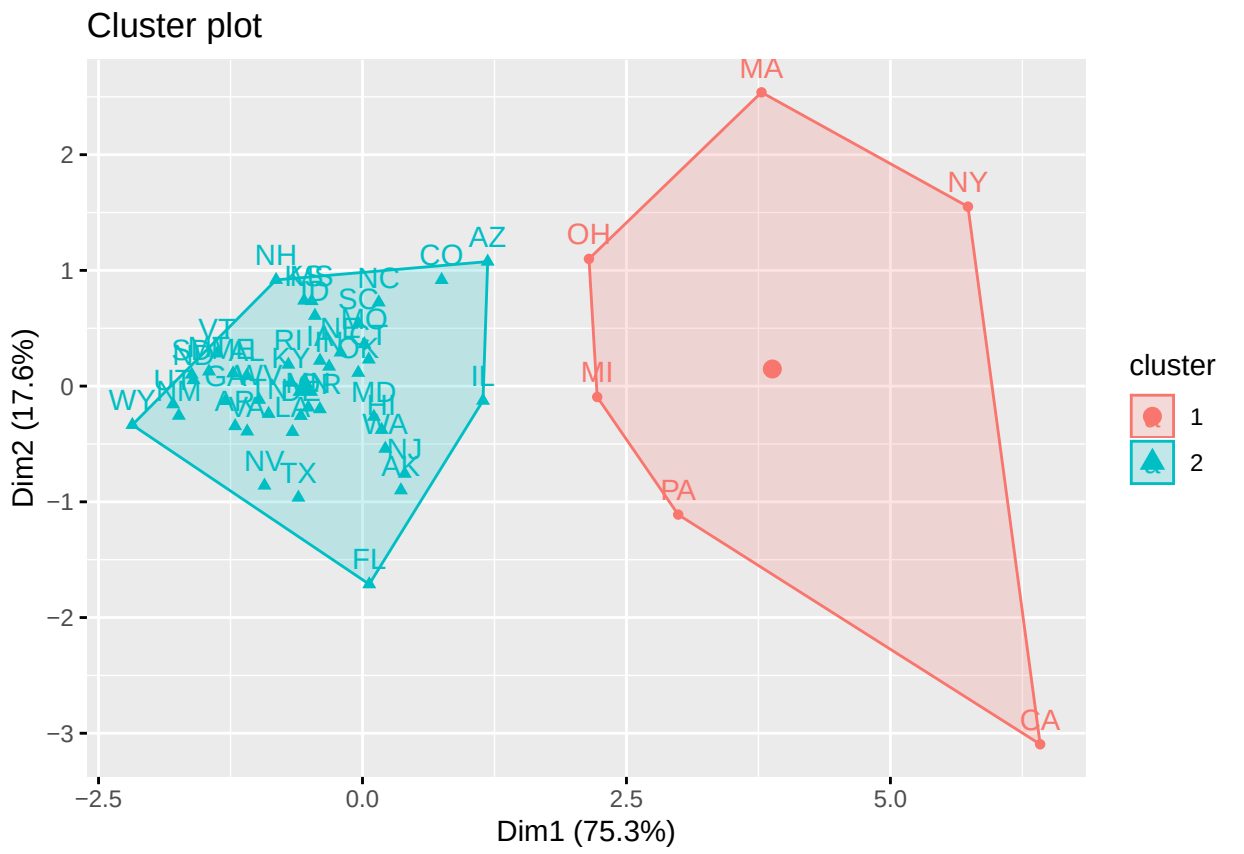
```
df_tree<- df%>%
  dist()%>%
  hclust(method = "complete")
plot(df_tree, cex=0.7, hang=-1)
```



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
```



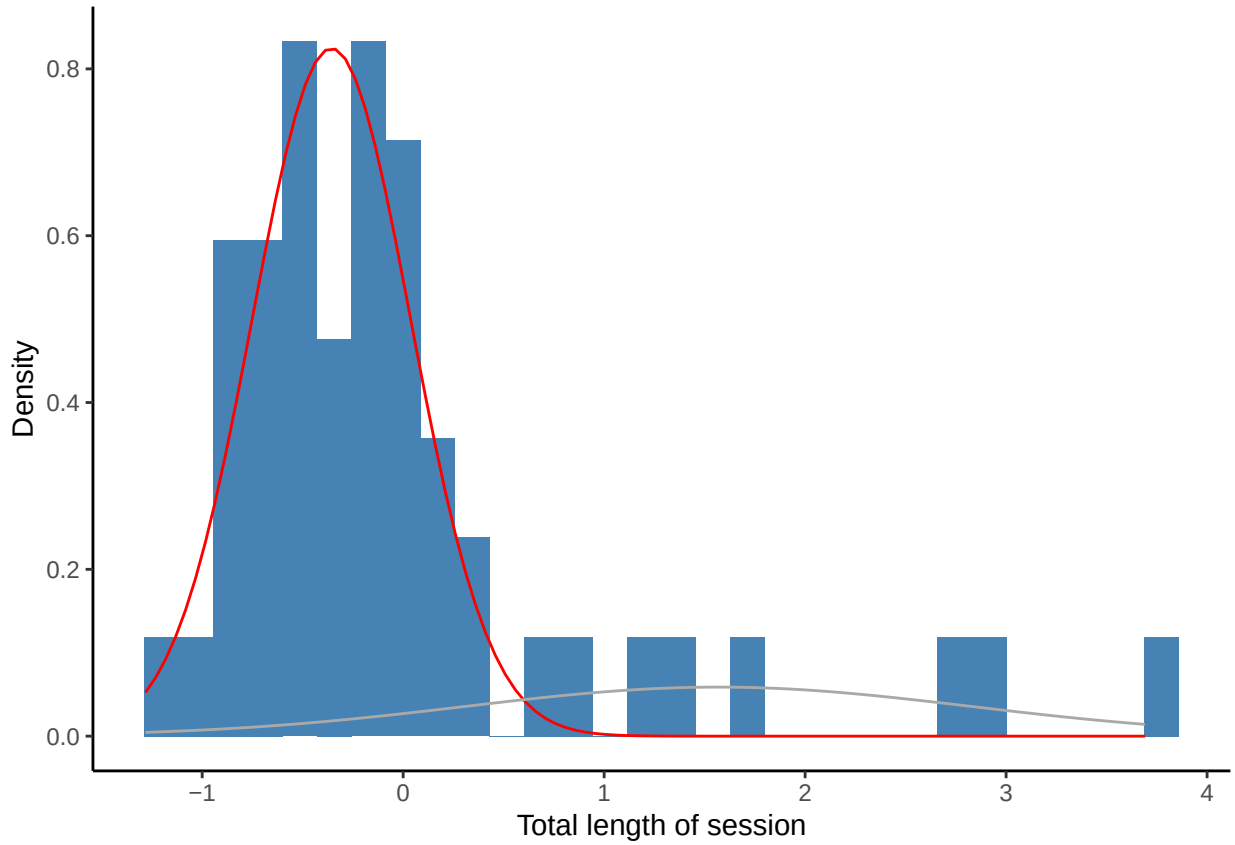
2.6

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.

```
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)
```

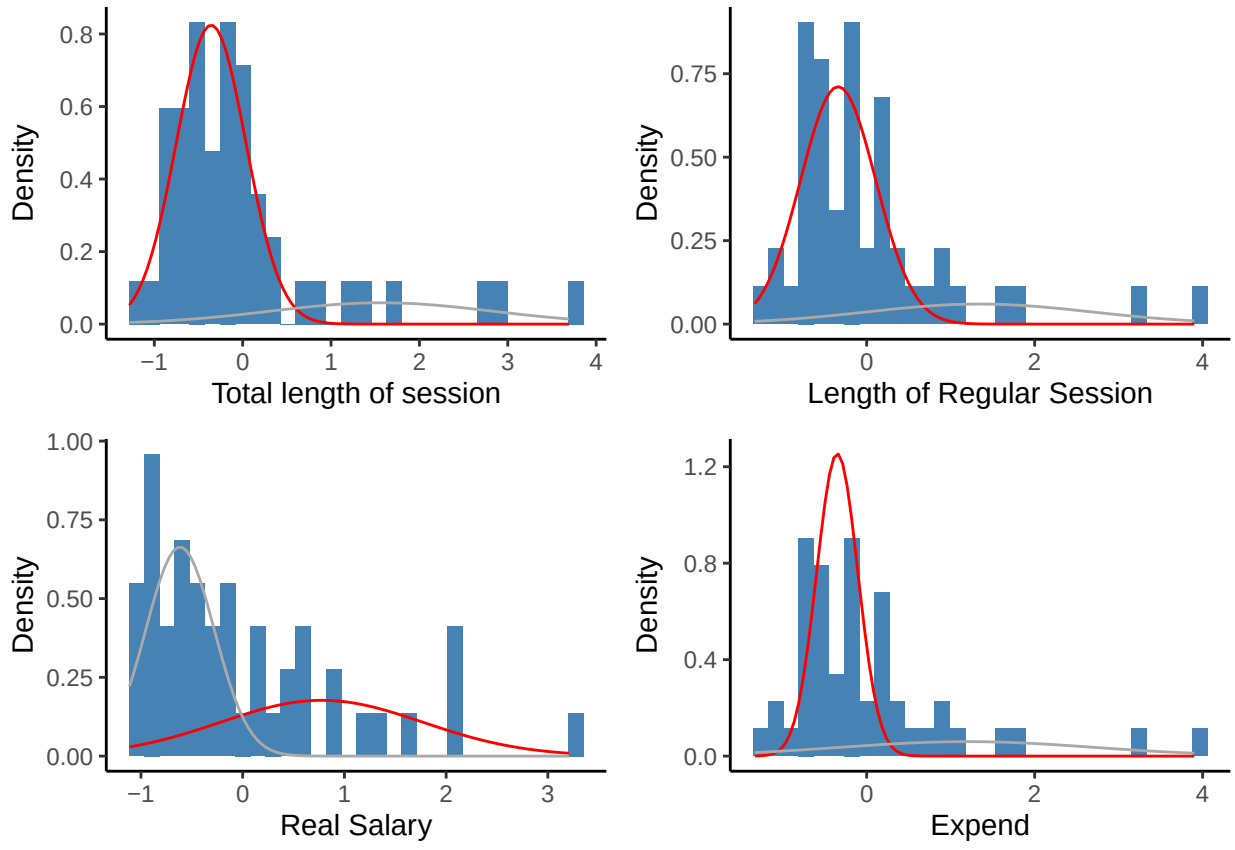
```
## number of iterations= 45
```

```
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

{r} valid<- clValid(df, 2:5, clMethods = c(“hierachical”, “kmeans”, “model”), validation= “internal”)

2.9

Basically, different internal validation methods choose different optimization algorithm.

It is really hard to determine which approach is optimal, which requires further literature analysis. However, as for the optimal k in the hierachical method, the value of the k is supposed to be 2, with which we find one large and one small cluster.

It is highly likely that we choose a sub-optimal approach based on different context.