Homework 4

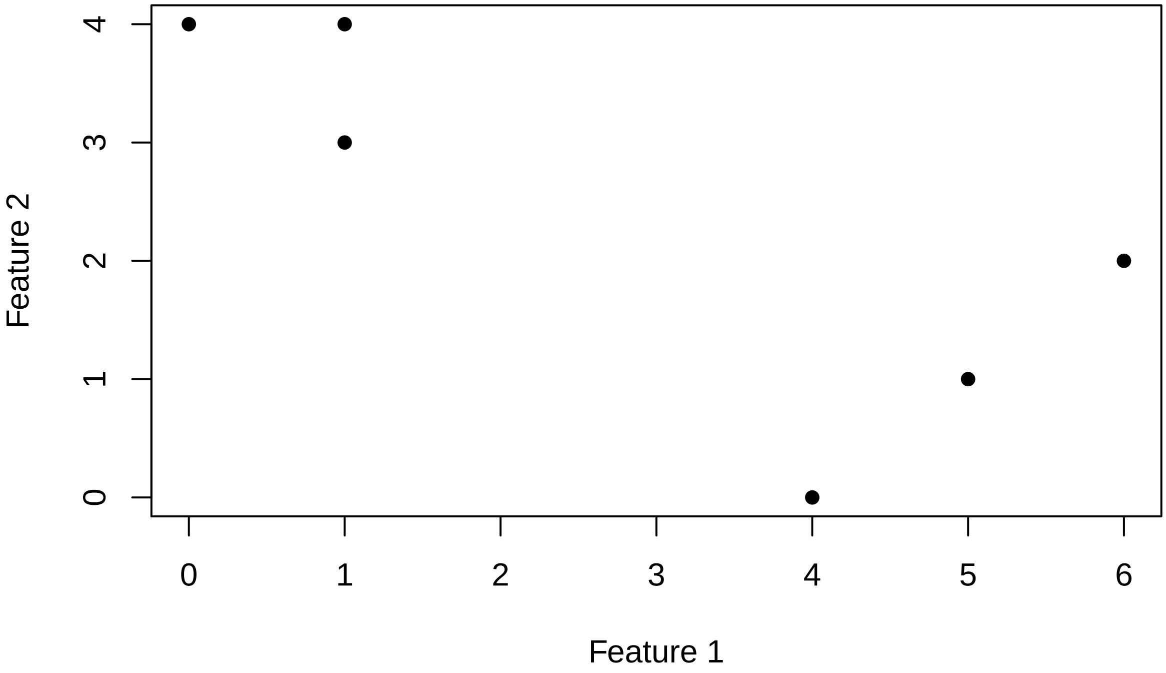
Michael Carrion

March 2, 2020

Performing k-Means by Hand

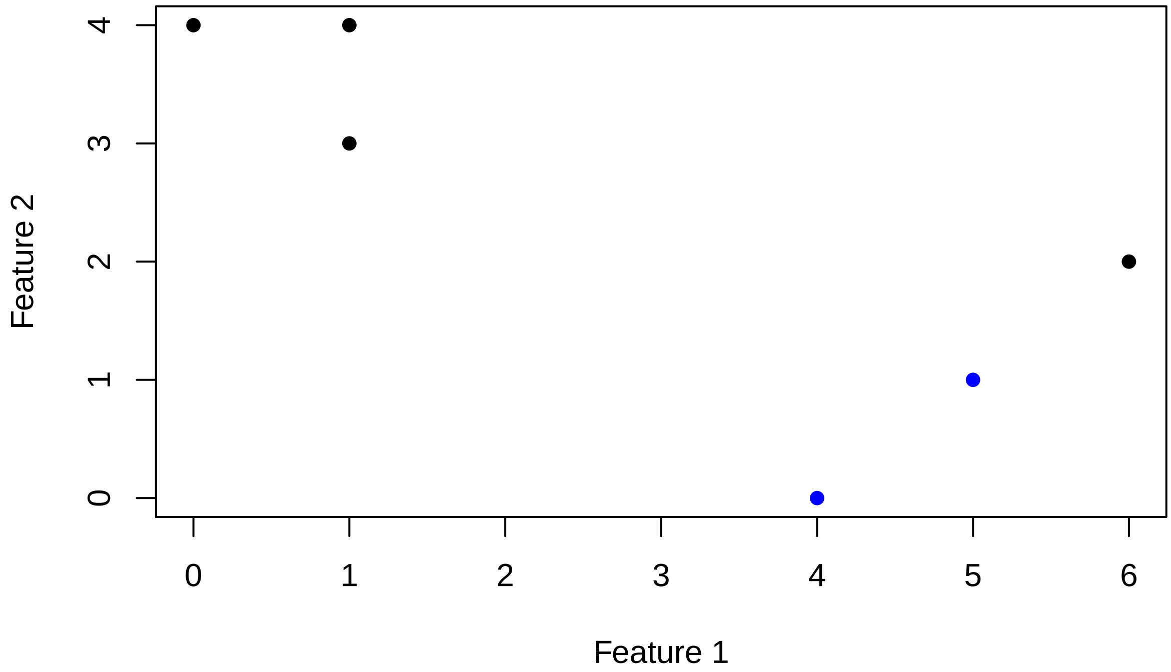
1.

```
x <- cbind(c(1, 1, 0, 5, 6, 4), c(4, 3, 4, 1, 2, 0))
k <- 2
plot(x,xlab="Feature 1",ylab="Feature 2",pch=16)
```



2.

```
set.seed(123)
cluster <- sample(seq(0,1), size=6, replace=TRUE)
xMod <- cbind(x,cluster)
plot(x[,1],x[,2],col=rgb(0,0,cluster),pch=16,xlab="Feature 1",ylab="Feature 2")
```



3.

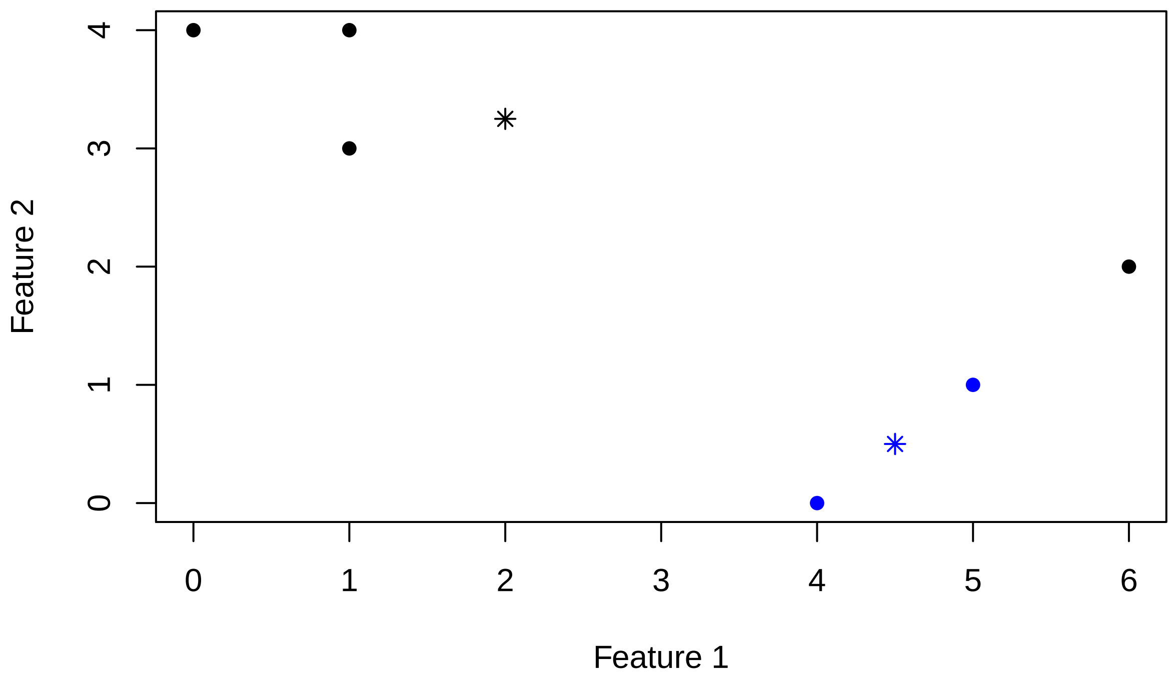
```
x0 <- mean(xMod[,1][xMod[,3]==0])
y0 <- mean(xMod[,2][xMod[,3]==0])

x1 <- mean(xMod[,1][xMod[,3]==1])
y1 <- mean(xMod[,2][xMod[,3]==1])

c(x0,y0) #Centroid for cluster 0
```

```
## [1] 2.00 3.25
```

```
c(x1,y1) #Centroid for cluster 1
```

```
## [1] 4.5 0.5
```

```
plot(x[,1],x[,2],col=rgb(0,0,cluster),pch=16,xlab="Feature 1",ylab="Feature 2")
points(x0,y0,col="black",pch=8)
points(x1,y1,col="blue",pch=8)
```



Note that the black and blue $*$ denote the centroid for the two clusters.

4.

```
myDist <- function(x1, x2) sqrt(sum((x1 - x2) ^ 2))
out <- NULL
c0 <- c(x0,y0)
c1 <- c(x1,y1)
for(i in 1:nrow(x)){
    out[i] <- if (myDist(x[i,],c0) <= myDist(x[i,],c1)) 0 else 1
}
xMod2 <- cbind(x,out)
xMod2
```

```
##           out
## [1,] 1 4   0
## [2,] 1 3   0
## [3,] 0 4   0
## [4,] 5 1   1
## [5,] 6 2   1
## [6,] 4 0   1
```

Thus, we see the fifth observation was relabled from cluster 0 to cluster 1.

5.

```
x0 <- mean(xMod2[,1][xMod2[,3]==0])
y0 <- mean(xMod2[,2][xMod2[,3]==0])

x1 <- mean(xMod2[,1][xMod2[,3]==1])
y1 <- mean(xMod2[,2][xMod2[,3]==1])

out <- NULL
c0 <- c(x0,y0)
c1 <- c(x1,y1)
```

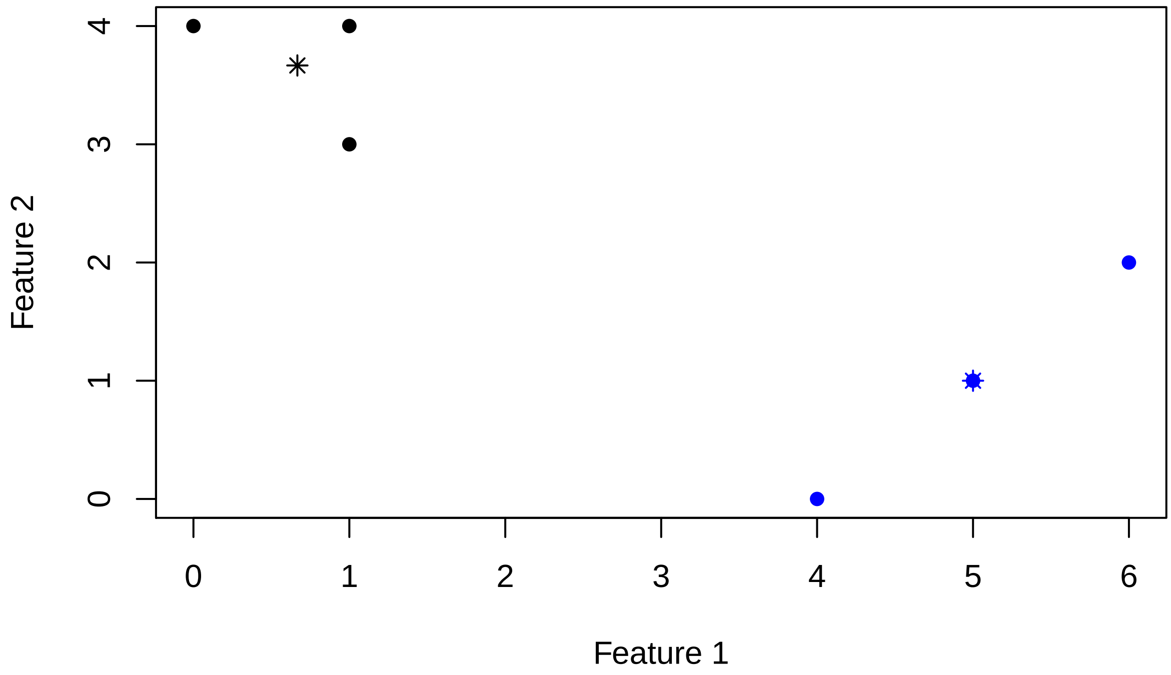
```
for(i in 1:nrow(x)){
    out[i] <- if (myDist(x[i,],c0) <= myDist(x[i,],c1)) 0 else 1
}
xMod3 <- cbind(x,out)
xMod3
```

```
##          out
## [1,] 1 4   0
## [2,] 1 3   0
## [3,] 0 4   0
## [4,] 5 1   1
## [5,] 6 2   1
## [6,] 4 0   1
```

Thus, we see that the the labels to the clusters stop changing after our first re-labeling (in Part 4).

6.

```
plot(x[,1],x[,2],col=rgb(0,0,out),pch=16,xlab="Feature 1",ylab="Feature 2")
points(x0,y0,col="black",pch=8)
points(x1,y1,col="blue",pch=8)
```



Note: As above, $*$ denotes the cluster centroids.

Clustering State Legislative Professionalism

1.

```
load("~/problem-set-4/Data and Codebook/legprof-components.v1.0.RData")
myDf <- x
```

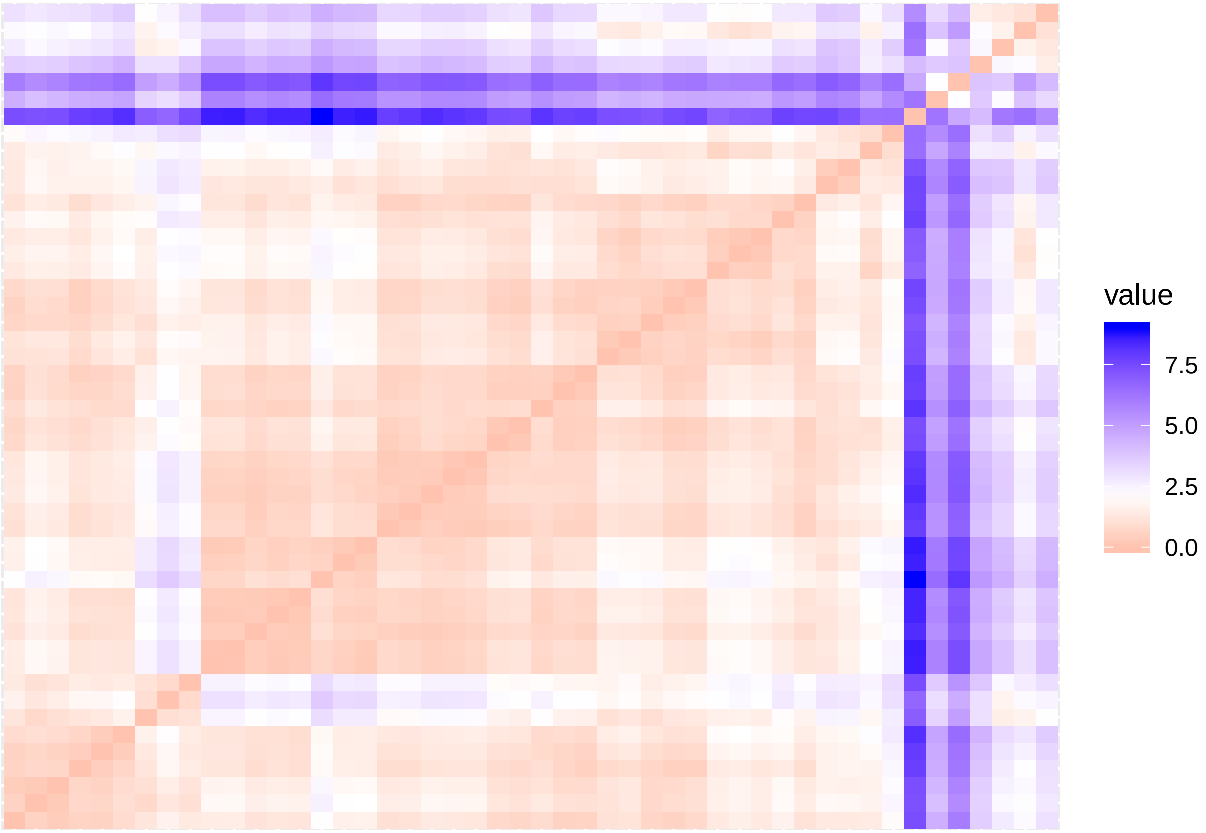
2.

```
myDfMod <- na.omit(myDf[myDf$year == "2009" | myDf$year == "2010",]) #b, #c
myDfSub <- myDfMod %>%
  select(t_slength, slength, salary_real, expend) %>% #a
  scale() #d
rownames(myDfSub) <- myDfMod$state #e, associated state names
```

3.

```
get_clust_tendency(myDfSub,45)
```

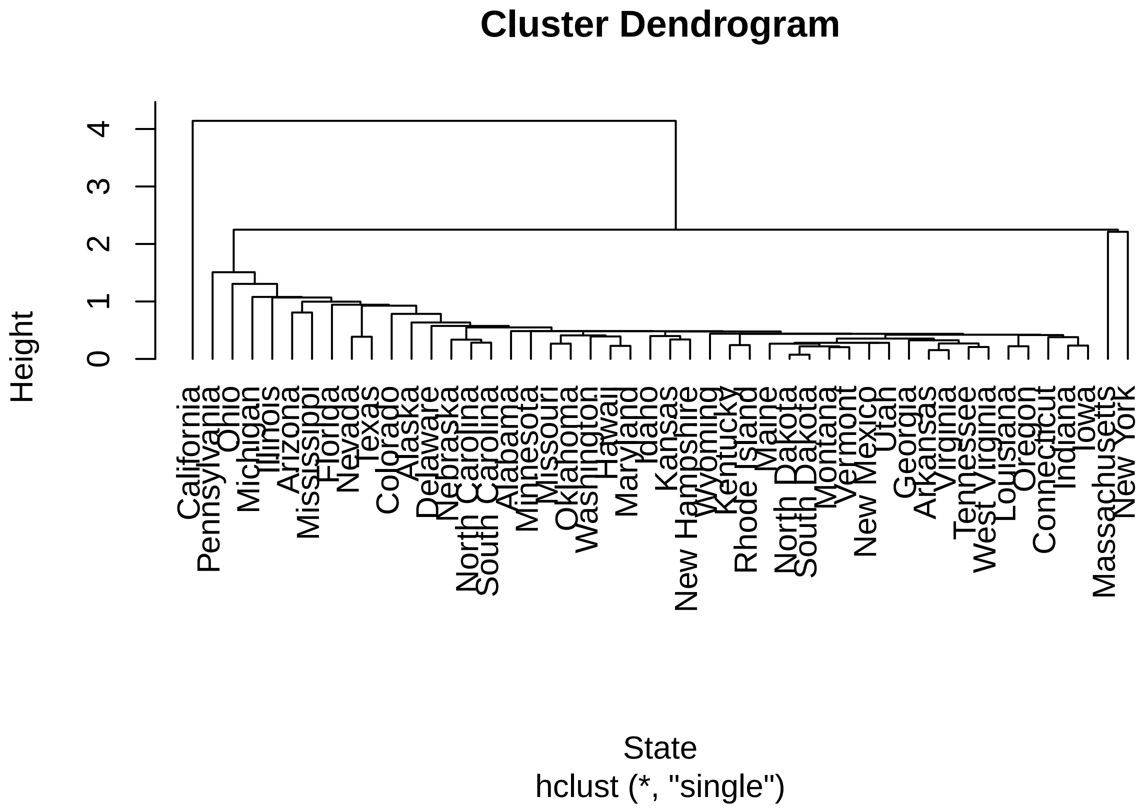
```
## $hopkins_stat
## [1] 0.8226008
##
## $plot
```



Aside from the first few observations, there are a lot of contiguous red & white squares in the dissimilarity image, suggesting high similarity (and thus clusterability) between certain observations. Also note that the Hopkins stat of 0.82 is close to 1, further suggesting clusterability.

4.

```
set.seed(825)
hc_single <- hclust(stats::dist(myDfSub), method = "single") #single linkage
plot(hc_single, hang = -1,xlab="State")
```



Note that the dendrogram above uses single linkage (the minimal inter-cluster dissimilarity). We see that California, Massachusetts, and New York are very dissimilar from the other clusters. There are a few patterns we'd expect (i.e. North and South Dakota are very similar, as are North and South Carolina), but it's hard to identify broader geographical patterns among the clusters (i.e. the Carolinas are most similar to Nebraska, Rhode Island is most similar to Kentucky, Hawaii is most similar to Maryland, etc.).

5.

```
set.seed(107)
kmeans <- kmeans(myDfSub, centers = 2, nstart = 35)
#Output Results
t <- as.table(kmeans$cluster)
t <- data.frame(t)
cluster2 <- t[t$Freq == "2",]
cluster2
```

```
##              Var1 Freq
## 5      California    2
## 21 Massachusetts    2
## 22       Michigan    2
## 31       New York    2
## 34           Ohio    2
## 37   Pennsylvania    2
```

```
kmeans$centers
```

```
##    t_slength    slength salary_real     expend
## 1 -0.2868507 -0.2949065    -0.29189 -0.2092542
## 2  2.0079549  2.0643454     2.04323  1.4647791
```

```
kmeans$size
```

```
## [1] 42  6
```

We see the above six states are classified as cluster 2. The remaining 42 are classified as cluster 1. Cluster 2 states have a much higher level of "professionalism" (that is, a greater session length, salary, and expenditure amount). Cluster 1 states thus have shorter sessions, lower salaries, and lower expenditure amounts.

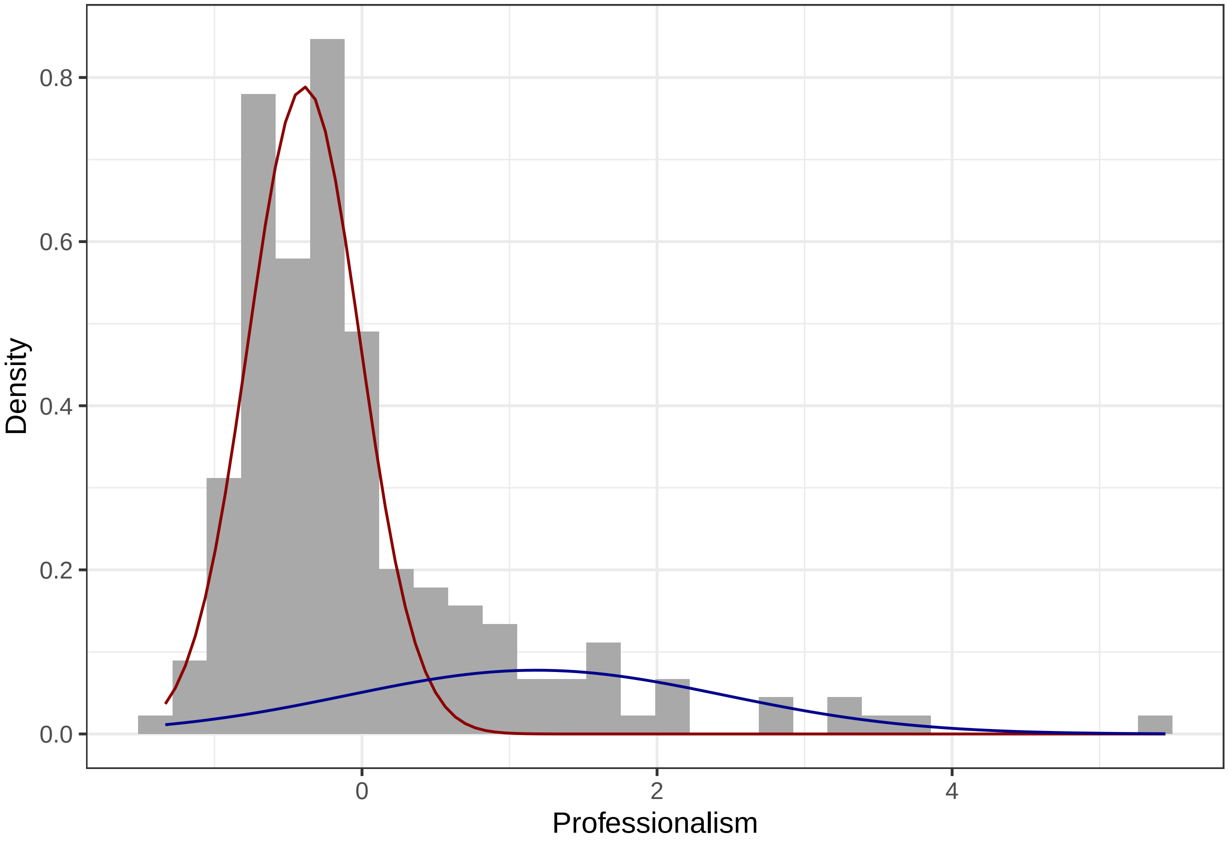
6.

```
set.seed(276)
gmm1 <- normalmixEM(myDfSub, k = 2)
```

```
## number of iterations= 37
```

```
#test <- myDfSub[-c(5,31),]  #Note: Removing the extreme values CA, NY did not change fig. greatly
ggplot(data.frame(x = gmm1$x)) +
  geom_histogram(aes(x, ..density..), fill = "darkgray") +
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm1$mu[1], gmm1$sigma[1], lam = gmm1$lambda[1]),
                colour = "darkred") +
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm1$mu[2], gmm1$sigma[2], lam = gmm1$lambda[2]),
                colour = "darkblue") +
  xlab("Professionalism") +
  ylab("Density") +
  theme_bw()
```

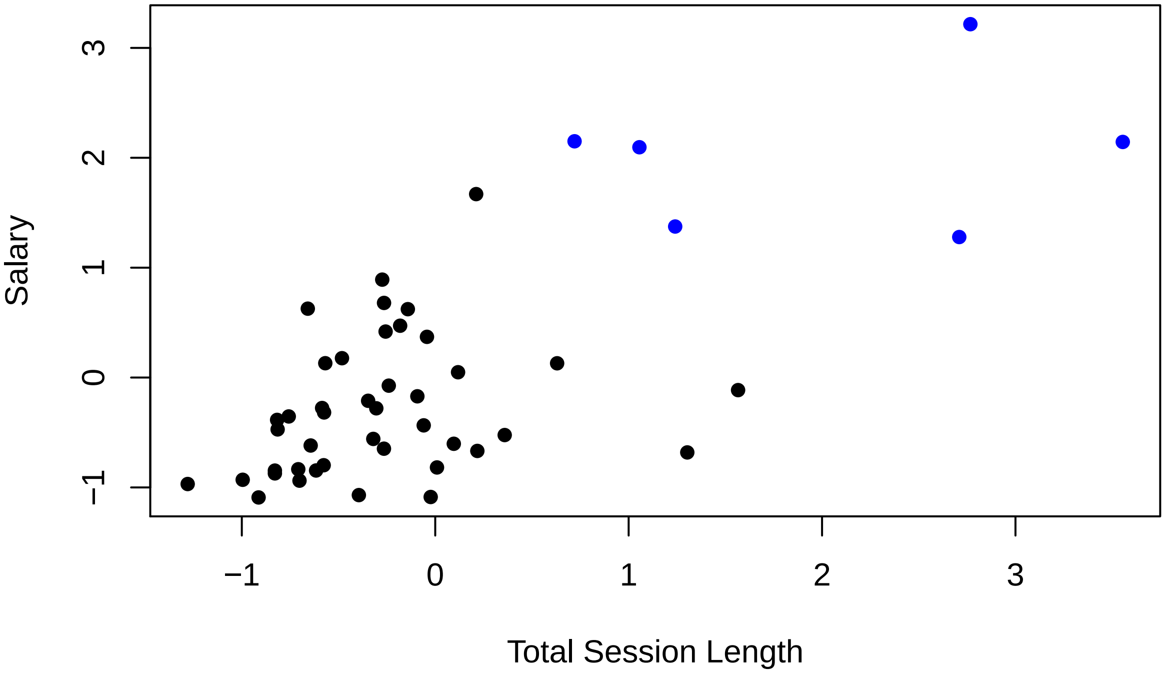
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



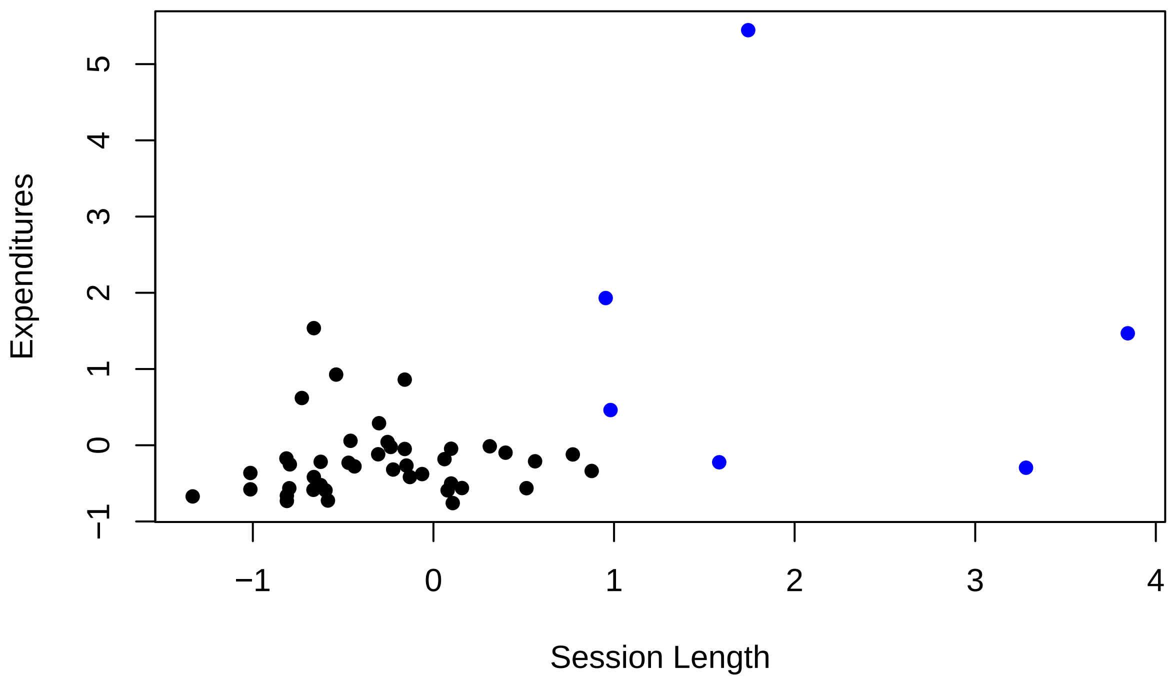
From the figure above, we see the large number of states in cluster 1, with the centroid slightly below 0. A smaller number of states are in cluster 2, which have professionalism values much more spread out above 1. Note that removing the apparent outlier (professionalism = ~6) did not significantly change the figure.

7.

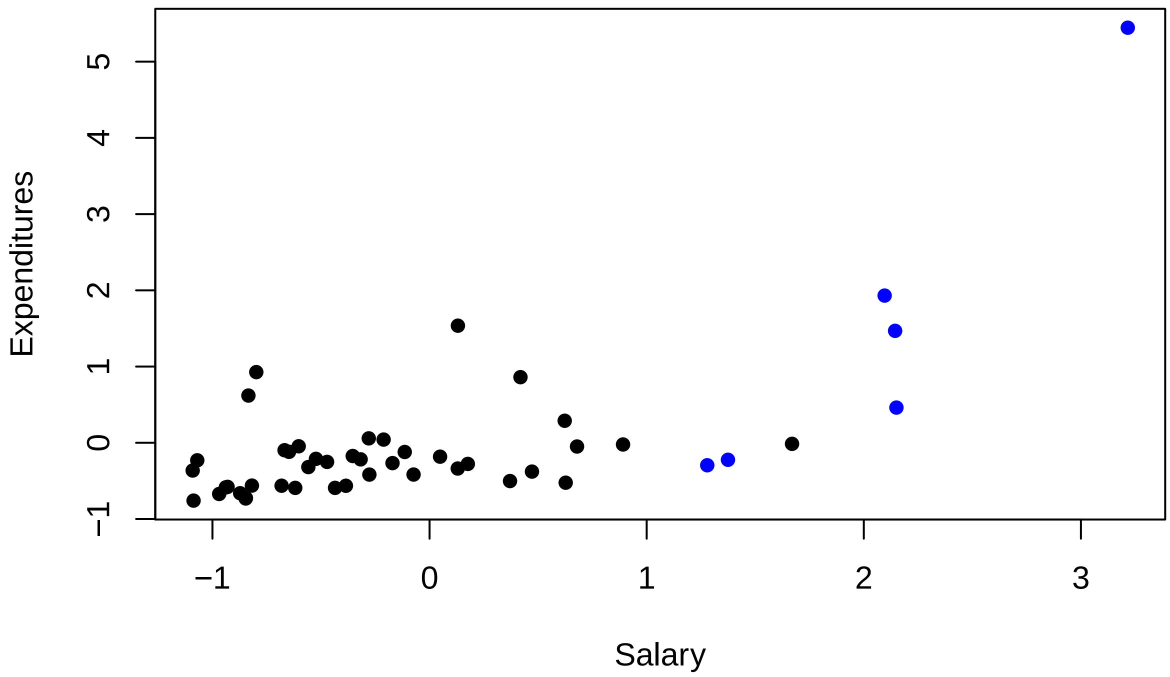
```
#Tidy up k-means assignments
rownames(t) <- myDfMod$state
colnames(t)[colnames(t)=="Freq"] <- "Assignment"
t$Var1 <- NULL
#add cluster assignments to df
merged <- merge(myDfSub,t,by.x = 0, by.y = 0)
merged["Assignment"] = merged["Assignment"]-1
plot(merged[,2],merged[,4],col=rgb(0,0,merged[,"Assignment"]),pch=16,
     xlab="Total Session Length",ylab="Salary")
```



```
plot(merged[,3],merged[,5],col=rgb(0,0,merged[,"Assignment"]),pch=16,
     xlab="Session Length",ylab="Expenditures")
```



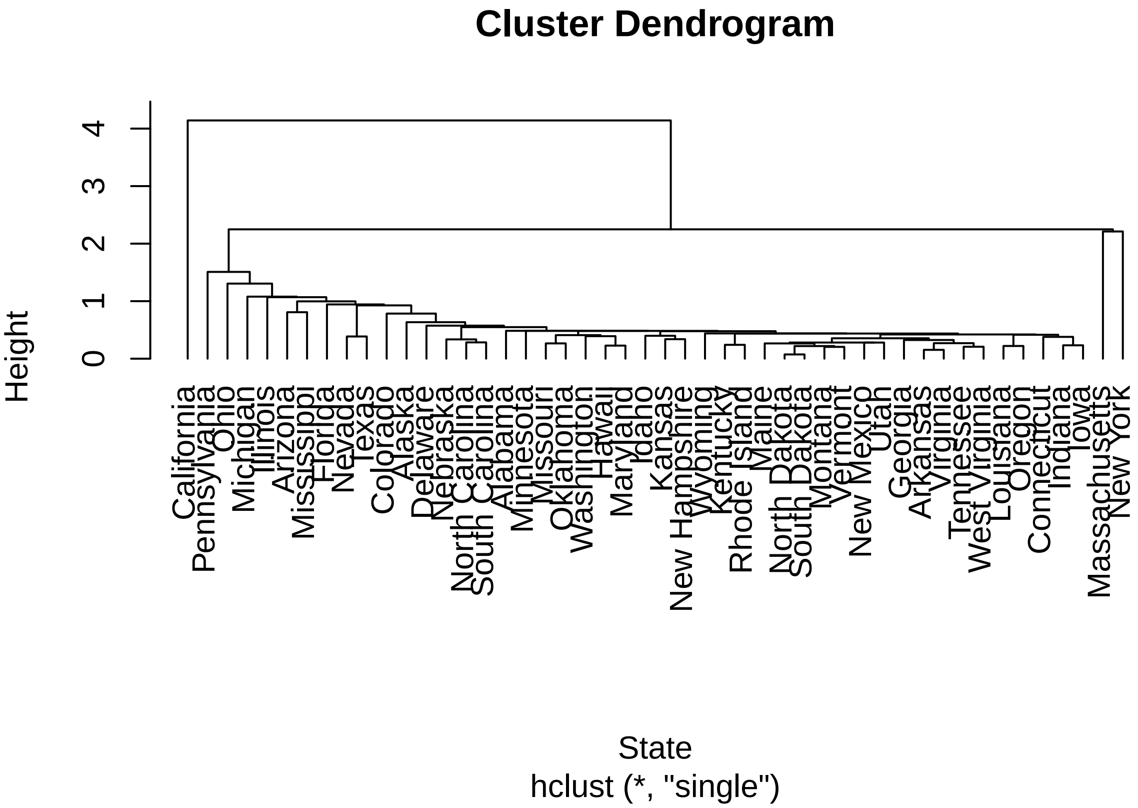
```
plot(merged[,4],merged[,5],col=rgb(0,0,merged[,"Assignment"]),pch=16,
     xlab="Salary",ylab="Expenditures")
```



Thus, from the scatter plots above, we see that our k-means model did a relatively good job of clustering the observations based on a variety of features.

Recall the dendrogram from (4) and the cluster classifications from our k-means model.

```r
plot(hc_single, hang = -1, xlab="State")
```



```r
cluster2
```

```
##             Var1 Freq
## 5     California    2
## 21 Massachusetts    2
## 22      Michigan    2
## 31      New York    2
## 34          Ohio    2
## 37  Pennsylvania    2
```

Thus, we can see that the most dissimilar states were all classified as cluster 2, as we'd expect.

Next, recall the centroids for our k-means model and our Gaussian mixture model

```r
kmeans$center
```

```
##    t_slength    slength salary_real     expend
## 1 -0.2868507 -0.2949065    -0.29189 -0.2092542
## 2  2.0079549  2.0643454     2.04323  1.4647791
```

```r
gmm1$mu
```

```
## [1] -0.3919016  1.1807307
```

Thus, we can see that in both models the centroid for the first cluster is negative, at about -0.3, and the centroid for our second cluster is positive, at about 2 in our k-means model and at about 1.2 in our gmm model.

Further, as previously noted, though it's tough to display the component densities because we have many components, from the plot in (6) it is apparent that our gmm model classifies most observations as cluster 1, with significantly fewer belong to cluster 2.

Thus, our three models largely seem to coincide in (1) the number of states belonging to each cluster, (2) which specific states belong to each cluster, and (3) the centroid/average values of each cluster.

8.

```
rownames(myDfSub) <- 1:nrow(myDfSub) #change names back
valid <- clValid(myDfSub, c(2), clMethods = c("hierarchical", "kmeans","model"),
                 validation = c("internal"))
summary(valid)
```

```
##
## Clustering Methods:
##  hierarchical kmeans model
##
## Cluster sizes:
##  2
##
## Validation Measures:
##                                  2
##
## hierarchical Connectivity   6.0869
##              Dunn           0.3598
##              Silhouette     0.6920
## kmeans       Connectivity   8.5683
##              Dunn           0.1726
##              Silhouette     0.6390
## model        Connectivity  18.7095
##              Dunn           0.0833
##              Silhouette     0.4230
##
## Optimal Scores:
##
##              Score  Method       Clusters
## Connectivity 6.0869 hierarchical 2
## Dunn         0.3598 hierarchical 2
## Silhouette   0.6920 hierarchical 2
```

Thus, for the hierarchical, k-means, and gmm models above (that is, using $k = 2$), we obtained silhouette scores of 0.69, 0.64, and 0.42, respectively. This suggests that the heirarchical model (with single linkage) performed the best, followed by the k-means model. Further, iterating over several starting values of k (namely from $k = 2:10,\ 20$), we see that the hierarchical model with $k = 2$ as initially proposed is still optimal using our silhouette-width metric. Thus, our assumption of using $k = 2$ in questions (5) and (6) above is justified.

```
valid2 <- clValid(myDfSub, c(2:10,20), clMethods = c("hierarchical", "kmeans","model"),
                 validation = c("internal"))
summary(valid2)
```

```
##
## Clustering Methods:
##  hierarchical kmeans model
##
## Cluster sizes:
##  2 3 4 5 6 7 8 9 10 20
##
```

```
## Validation Measures:
##                                  2       3       4       5       6       7       8       9      10
##
## hierarchical Connectivity   6.0869  6.9536 13.1345 15.1345 20.7563 22.9230 28.1726 30.1171 40.5512 7
##              Dunn           0.3598  0.4340  0.2902  0.2902  0.2836  0.2836  0.2451  0.2451  0.1930  0
##              Silhouette     0.6920  0.6619  0.5199  0.4989  0.3776  0.3658  0.2921  0.2831  0.2624  0
## kmeans       Connectivity   8.5683 11.0183 18.1651 20.1651 23.6810 25.8476 36.4726 44.8750 45.4024 77
##              Dunn           0.1726  0.2597  0.2456  0.2456  0.1214  0.1214  0.1871  0.1846  0.2515  0
##              Silhouette     0.6390  0.6054  0.4824  0.4611  0.3328  0.3210  0.3169  0.2854  0.3249  0
## model        Connectivity  18.7095 23.7964 33.3683 60.1651 69.0651 54.4433 51.8206 63.7619 62.1766 77
##              Dunn           0.0833  0.0855  0.0554  0.0280  0.0391  0.0532  0.0935  0.0879  0.0928  0
##              Silhouette     0.4230  0.3854  0.2157  0.0962  0.0473  0.1822  0.2957  0.2091  0.2132  0
##
## Optimal Scores:
##
##              Score  Method       Clusters
## Connectivity 6.0869 hierarchical 2
## Dunn         0.4340 hierarchical 3
## Silhouette   0.6920 hierarchical 2
```

9.

As stated above, the hierarchical model with $k = 2$ clusters performed best using our silhouette-width metric. However, under the Dunn Index metric, the hierarchical model with 3 clusters performed best, so further examination into this model should be conducted before definitively selecting one over the other. Further, a silhouette value close to 1 is ideal, so our optimal value of 0.69 suggests a model with fair accuracy.

There are several reasons why one might select a “sub-optimal” clustering method. For example, one can imagine selecting the same amount of clusters as data points. This would result in a model with perfect accuracy on the training set (no bias), but extremely high variance, and thus would not be ideal. That is, if you include too many clusters, you may begin to model the random noise in the data. Thus, even though a model with more clusters might result in reduced error, it may not necessarily be the optimal choice. Another reason might concern interpretability. For example, when considering political preferences, it might be better to try to classify someone as a republican or democrat (two clusters), even though these clusters may not perfectly capture his/her preferences.