Tammy Glazer Problem Set 3

10/25/2019

Question 1

Load the state legislative professionalism data from the folder. See the codebook for reference in the same folder and combine with our discussion of these data and the concept of state legislative professionalism from class for relevant background information.

```
library(tidyverse)
library(grid)
library(gridExtra)
library(knitr)
library(seriation)
library(ggpubr)
library(factoextra)
library(clValid)
library(mixtools)
library(plotGMM)
library(factoextra)
library(mclust)
library(fpc)

load('legprof-components.v1.0.RData')
data <- x
```

Question 2

Munge the data.

a. Select only the continuous features that should capture a state legislature's level of "professionalism" (session length (total and regular), salary, and expenditures)

b. Restrict the data to only include the 2009/10 legislative session for consistency

c. Omit all missing values

d. Standardize the input features

e. And anything else you think necessary to get this subset of data into workable form (hint: consider storing the state names as a separate object to be used in plotting later)

```
sub <- data %>%
       select(state, sessid, t_slength, slength, salary_real, expend) %>%
       filter(sessid == "2009/10") %>%
       drop_na() %>%
       mutate_if(is.numeric, funs(scale)) %>%
```

```
       select(-sessid)

state_names <- sub[,1]
data_final <- sub %>% select(-state)
```

Question 3

Perform quick EDA visually or numerically and discuss the patterns you see.

```
summary(data_final)
```

```
##     t_slength.V1         slength.V1        salary_real.V1
##  Min.   :-1.282138   Min.   :-1.331915   Min.   :-1.113266
##  1st Qu.:-0.599190   1st Qu.:-0.615579   1st Qu.:-0.714573
##  Median :-0.238210   Median :-0.210107   Median :-0.296849
##  Mean   : 0.000000   Mean   : 0.000000   Mean   : 0.000000
##  3rd Qu.: 0.133236   3rd Qu.: 0.171443   3rd Qu.: 0.454255
##  Max.   : 3.691295   Max.   : 3.900711   Max.   : 3.206991
##       expend.V1
##  Min.   :-0.772770
##  1st Qu.:-0.535853
##  Median :-0.239991
##  Mean   : 0.000000
##  3rd Qu.:-0.022427
##  Max.   : 5.478545
```

Given that the continuous variables are standardized, they all have a mean of zero and a standard deviation of 1. However, it is interesting to note that session length, total session length, salary, and expenditures have medians below the mean, indicating a potential skew in the data toward lower values. In relative terms, session length has the lowest value in its range and expenditures has the highest value in its range.

```
nrow(data_final)
```

```
## [1] 49
```

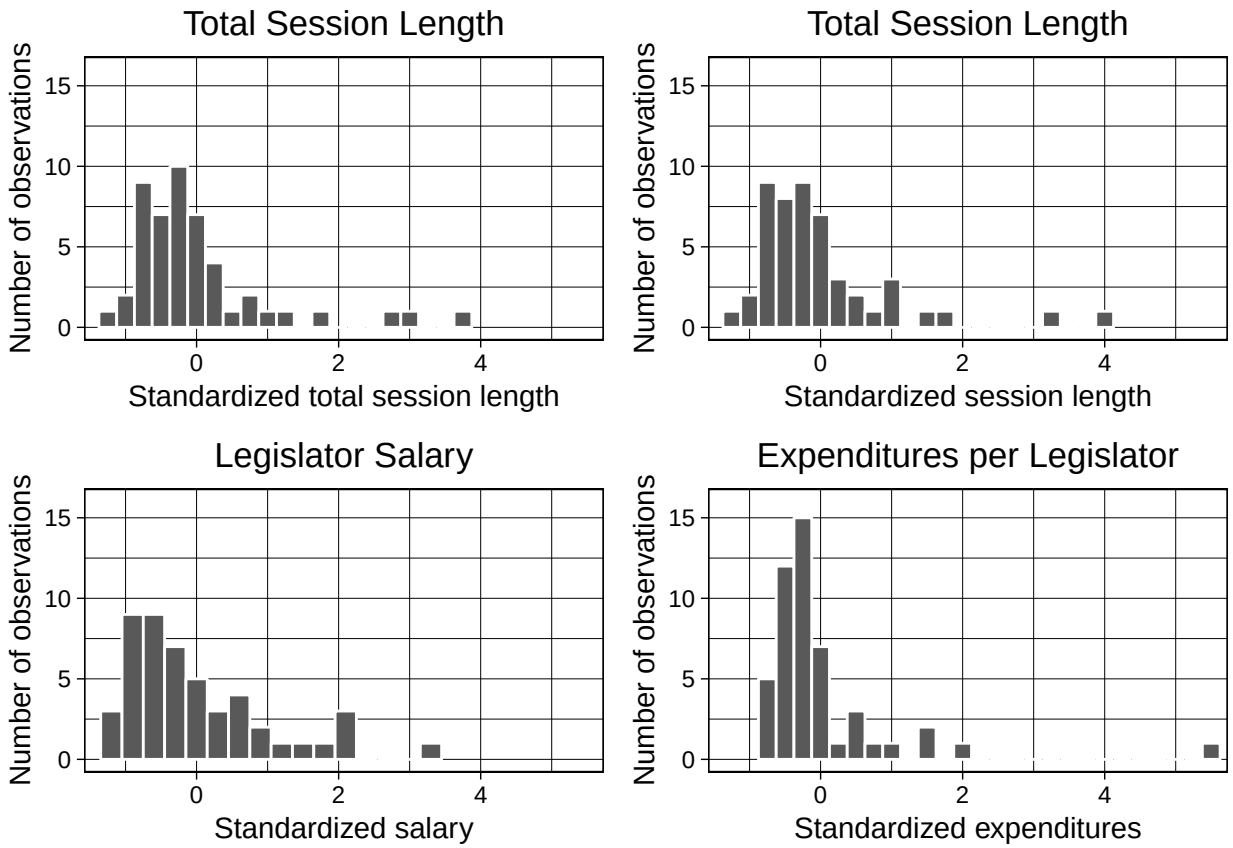
There are data for 49 states in this set, indicating that one state was dropped for having an NA value.

```
a <- IQR(data_final$t_slength)
b <- IQR(data_final$slength)
c <- IQR(data_final$salary_real)
d <- IQR(data_final$expend)

kable(data.frame("Variable" = c("t_slength", "slength", "salary_real", "expend"),
                 "IQR" = c(a, b, c, d)))
```

Variable	IQR
t_slength	0.7324261
slength	0.7870225
salary_real	1.1688275
expend	0.5134260

Interquartile range is a measure of spread with a high resistance to outliers. Salary has the highest interquartile range of the continuous variables of interest while expenditures has the lowest interquartile range of the coninuous variables of interest.

```
plot1 <- ggplot(data_final) +
         geom_histogram(aes(x = t_slength), binwidth = 0.25, color = "white") +
         labs(x = "Standardized total session length",
              y = "Number of observations") +
         ggtitle("Total Session Length") +
         theme_linedraw() +
         coord_cartesian(xlim =c(-1.25, 5.4), ylim =c(0, 16)) +
         theme(plot.title = element_text(hjust = 0.5))

plot2 <- ggplot(data_final) +
         geom_histogram(aes(x = slength), binwidth = 0.25, color = "white") +
         labs(x = "Standardized session length",
              y = "Number of observations") +
         ggtitle("Total Session Length") +
         theme_linedraw() +
         coord_cartesian(xlim =c(-1.25, 5.4), ylim =c(0, 16)) +
         theme(plot.title = element_text(hjust = 0.5))

plot3 <- ggplot(data_final) +
         geom_histogram(aes(x = salary_real), binwidth = 0.3, color = "white") +
         labs(x = "Standardized salary",
              y = "Number of observations") +
         ggtitle("Legislator Salary") +
         theme_linedraw() +
         coord_cartesian(xlim =c(-1.25, 5.4), ylim =c(0, 16)) +
         theme(plot.title = element_text(hjust = 0.5))

plot4 <- ggplot(data_final) +
         geom_histogram(aes(x = expend), binwidth = 0.25, color = "white") +
         labs(x = "Standardized expenditures",
              y = "Number of observations") +
         ggtitle("Expenditures per Legislator") +
         theme_linedraw() +
         coord_cartesian(xlim =c(-1.25, 5.4), ylim =c(0, 16)) +
         theme(plot.title = element_text(hjust = 0.5))

grid.arrange(plot1, plot2, plot3, plot4, ncol=2, nrow=2)
```

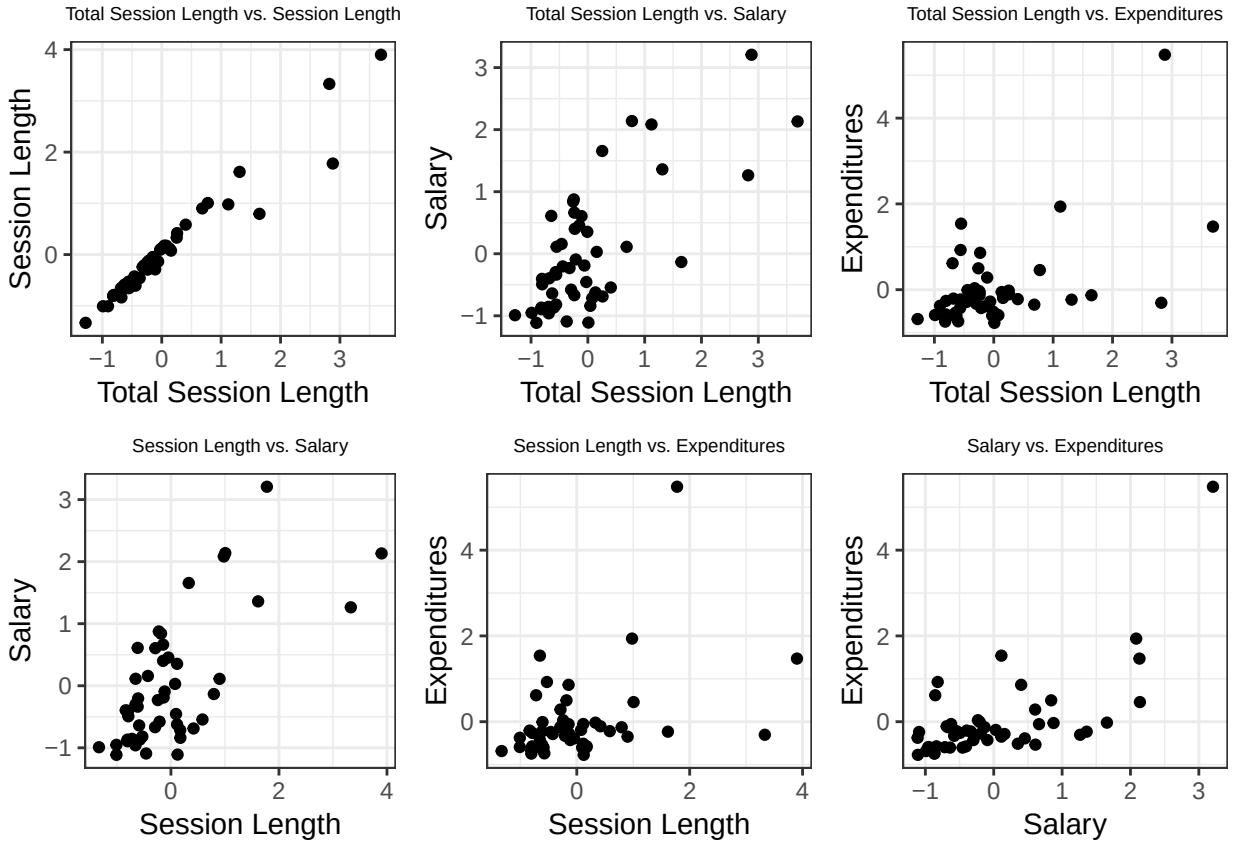


In this dataset, more than half of the observations fall below the mean (0) for all continuous variables. A consistent trend among the variables seems to be a steady increase in number of observations and a peak just lower than the mean, followed by a quick drop-off in number of observations for values greater than the mean. Total session length is the one variable that displays a drop in the number of observations below the mean followed by another spike before the mean. There are only a handful of observations above 0 for each variable.

Additionally, there consistently seem to be strong outliers at higher values for these continuous variables. For instance, for expenditures, there is a single observation just under 6; for salary, there is a single observation just after 3; for total session length, there are two observations above 3; and for total session length, there are 3 outliers greater than 2. These observations likely have high magnitudes in real-world terms, since the process of standardizing indicates that each variable has a standard deviation of 1.

```
a <- ggplot(data_final, aes(x = t_slength, y = slength)) +
     geom_point() +
     labs(x = "Total Session Length",
          y = "Session Length",
          title = "Total Session Length vs. Session Length") +
     theme_bw() +
     theme(plot.title = element_text(hjust = 0.5)) +
     theme(plot.title = element_text(size=7))

b <- ggplot(data_final, aes(x = t_slength, y = salary_real)) +
     geom_point() +
     labs(x = "Total Session Length",
          y = "Salary",
          title = "Total Session Length vs. Salary") +
     theme_bw() +
```

```
     theme(plot.title = element_text(hjust = 0.5)) +
     theme(plot.title = element_text(size=7))

c <- ggplot(data_final, aes(x = t_slength, y = expend)) +
     geom_point() +
     labs(x = "Total Session Length",
          y = "Expenditures",
          title = "Total Session Length vs. Expenditures") +
     theme_bw() +
     theme(plot.title = element_text(hjust = 0.5)) +
     theme(plot.title = element_text(size=7))

d <- ggplot(data_final, aes(x = slength, y = salary_real)) +
     geom_point() +
     labs(x = "Session Length",
          y = "Salary",
          title = "Session Length vs. Salary") +
     theme_bw() +
     theme(plot.title = element_text(hjust = 0.5)) +
     theme(plot.title = element_text(size=7))

e <- ggplot(data_final, aes(x = slength, y = expend)) +
     geom_point() +
     labs(x = "Session Length",
          y = "Expenditures",
          title = "Session Length vs. Expenditures") +
     theme_bw() +
     theme(plot.title = element_text(hjust = 0.5)) +
     theme(plot.title = element_text(size=7))

f <- ggplot(data_final, aes(x = salary_real, y = expend)) +
     geom_point() +
     labs(x = "Salary",
          y = "Expenditures",
          title = "Salary vs. Expenditures") +
     theme_bw() +
     theme(plot.title = element_text(hjust = 0.5)) +
     theme(plot.title = element_text(size=7))

grid.arrange(a, b, c, d, e, f, ncol=3, nrow=2)
```



Total session length and session length seem to have a linear, positive relationship, whereas most other combinations of variables do seem to have a positive relationship but not necessarily a linear one. In general, there seem to be apparent density-based clusters for all pairs of variables, with high density at low values and low density when the values of each increase. Again, each of the scatterplots have a handful of outliers at high values (approximately 6 for each).

Question 4

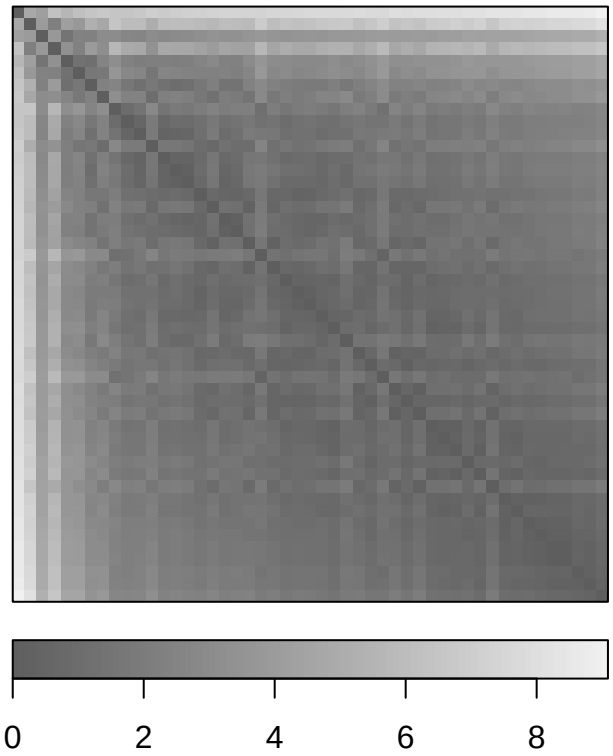
Diagnose clusterability in any way you'd prefer (eg. sparse sampling, ODI, etc.); Display the results and discuss the likelihood that natural, non-random structure exist in these data.

Method 1 - Informal techniques. Summarize measures of central tendancy and variation; Plot visualizations including a matrix of scatterplots.

Initial conclusion: Based on the above numerical and visual analysis, I have reason to believe that there may be natural, non-random structure in the data. For session length, total session length, salary, and expenditures, more than half of observations fall below the mean (0) and all variables display similar trends (increase in values before the mean, followed by a quick dropoff). Additionally, there seem to be strong outliers at higher values for all variables. The above scatterplots reveal a visual cue that there may be density-based clusters, with high density at low values and low density as the values of each variable increase. There are wide gaps creating distinct sections in the scatterplots.

Method 2 - Visual technique. VAT/DOI plot.

```
# Calculate a dissimilarity matrix of the data
dis_matrix <- data_final %>%
              dist(method="euclidean")

# Generate an ODI
dissplot(dis_matrix,
         main="ODI for Legislative Professionalism")
```



After ordering by spatial proximity, this ordered dissimilarity matrix (ODI) is less conclusive. Although it does not make a strong case for clusterability, it does demonstrate potential. There is a small dark square in the upper lefthand corner of the image as well as a very large dark square in the bottom righthand corder of the image, indicating groupings of similar vectors (low dissimilarity). The lighter sections on either side demonstrate high dissimilarity between these smaller groupings. That said, there are not clear boundaries around the sections, making it difficult to draw a conclusion with certainty.

Method 3 - Mathematical Technique. Hopkins Statistic.

```
stats <- get_clust_tendency(data_final, n=48, graph=FALSE)
stats$hopkins_stat
```

```
## [1] 0.1723366
```

A value of the Hopkins statistic close to 1 indicates that the data are highly clustered. A value close to 0 indicates a uniform distribution of the data. However, according to the documentation for get_clust_tendency, "if the value of Hopkins statistic is close to zero (far below 0.5), then we can conclude that the dataset is significantly clusterable" (1-Hopkins statistic). This low value, therefore, leads me to believe that I can reject the null hypothesis of spatial randomness in the data. Therefore, it does support a case for clusterability.

Question 5

Fit a k-means algorithm to these data and present the results. Give a quick, high level summary of the output and general patterns. Initialize the algorithm at k=2, and then check this assumption in the validation questions below.

```
set.seed(111)
data_final <- sub %>% select(-state)
kmeans <- kmeans(data_final,
                 centers = 2,
                 nstart = 15)
str(kmeans)
```

```
## List of 9
##  $ cluster     : int [1:49] 1 1 1 1 2 1 1 1 1 1 ...
##  $ centers     : num [1:2, 1:4] -0.293 2.1 -0.293 2.101 -0.283 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:4] "t_slength" "slength" "salary_real" "expend"
##  $ totss       : num 192
##  $ withinss    : num [1:2] 48.4 40.4
##  $ tot.withinss: num 88.7
##  $ betweenss   : num 103
##  $ size        : int [1:2] 43 6
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
kable(kmeans$centers)
```

t_slength	slength	salary_real	expend
-0.2930275	-0.2932285	-0.2833616	-0.2047966
2.1000302	2.1014710	2.0307585	1.4677087

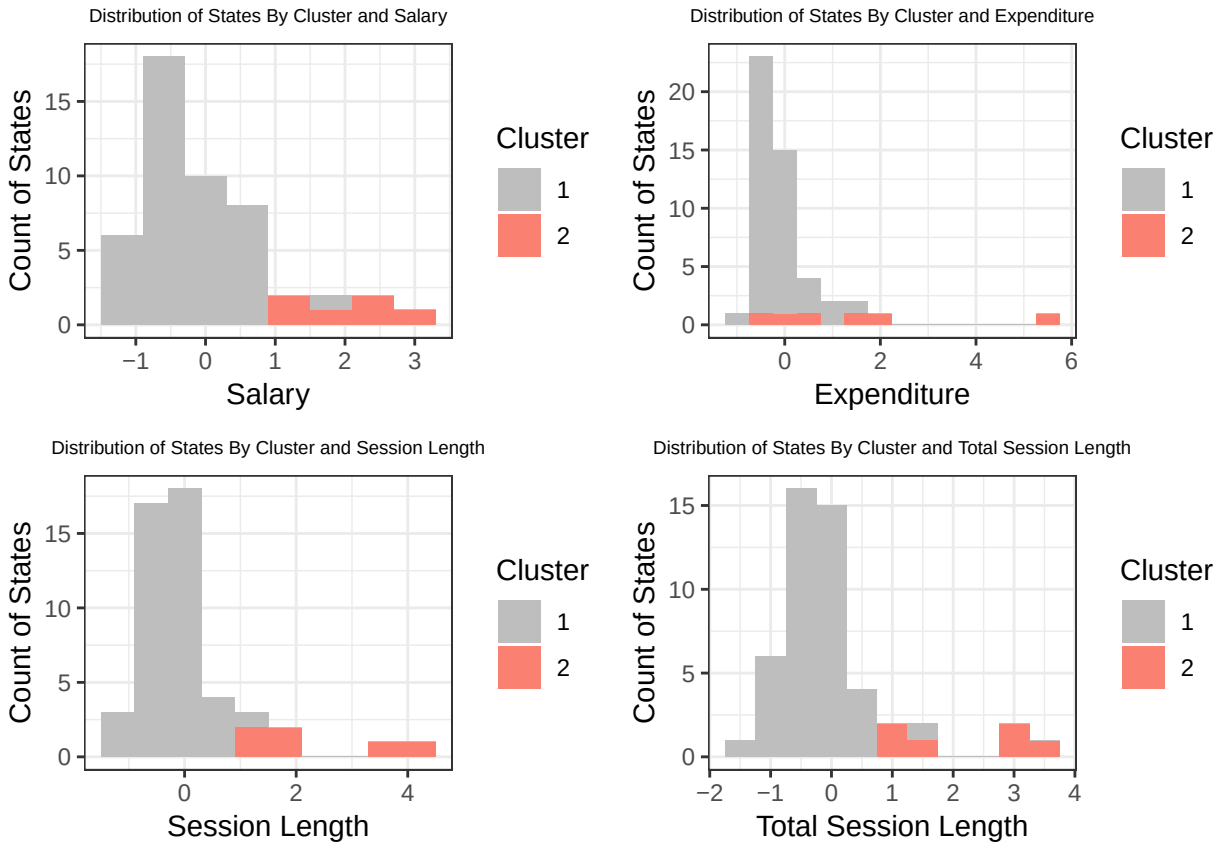
```
kmeans$size
```

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

```
data_final_k <- data_final
data_final_k$Cluster <- as.factor(kmeans$cluster)
t <- as.table(kmeans$cluster)
t <- data.frame(t)
rownames(t) <- state_names
colnames(t)[colnames(t) == "Freq"] <- "Assignment"
t$Var1 <- NULL

kable(t)
```

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

```r
# Distribution of states based on their cluster assignment
a <- ggplot(data_final_k, aes(salary_real, fill = Cluster)) +
    geom_histogram(binwidth = 0.6) +
    theme_bw() +
    scale_fill_manual(values=c("grey", "salmon")) +
    labs(x = "Salary",
         y = "Count of States",
         title = "Distribution of States By Cluster and Salary") +
    theme(plot.title = element_text(hjust = 0.5)) +
    theme(plot.title = element_text(size=7))

b <- ggplot(data_final_k, aes(expend, fill = Cluster)) +
    geom_histogram(binwidth = 0.5) +
    theme_bw() +
    scale_fill_manual(values=c("grey", "salmon")) +
    labs(x = "Expenditure",
         y = "Count of States",
         title = "Distribution of States By Cluster and Expenditure") +
    theme(plot.title = element_text(hjust = 0.5)) +
    theme(plot.title = element_text(size=7))

c <- ggplot(data_final_k, aes(slength, fill = Cluster)) +
    geom_histogram(binwidth = 0.6) +
    theme_bw() +
    scale_fill_manual(values=c("grey", "salmon")) +
    labs(x = "Session Length",
         y = "Count of States",
         title = "Distribution of States By Cluster and Session Length") +
    theme(plot.title = element_text(hjust = 0.5)) +
    theme(plot.title = element_text(size=7))

d <- ggplot(data_final_k, aes(t_slength, fill = Cluster)) +
    geom_histogram(binwidth = 0.5) +
    theme_bw() +
    scale_fill_manual(values=c("grey", "salmon")) +
    labs(x = "Total Session Length",
         y = "Count of States",
         title = "Distribution of States By Cluster and Total Session Length") +
    theme(plot.title = element_text(hjust = 0.5)) +
    theme(plot.title = element_text(size=7))

grid.arrange(a, b, c, d, ncol=2, nrow=2)
```



In k-means clustering, each cluster is represented by its center (centroid) which corresponds to the mean of points assigned to the cluster. In this case, the data have been separated into two clusters, with 6 states belonging to cluster 2 (California, Massachusetts, Michigan, New York, Ohio, Pennsylvania), and the the rest of the states belonging to cluster 1. For cluster 2 (with 6 states), the centroids are very high values (2.1 total session length, 2.1 session length, 2.03 salary, and 1.47 expenditures). For cluster 1 (with 43 states), the centroids are lower values close to the mean (-0.29 for total session length, -0.29 for session length, -0.28 for salary, and -0.20 for expenditures). The within cluster sum of squares is 40.4 for cluster 2 and 48.4 for cluster 1. By plotting the distribution of the four variables of interest based on cluster assignment, interesting patterns emerge. For salary, session length, and total session length, all of the states in cluster 2 appear to the left of cluster 1 (higher values), with a single state exception in each case. However, expenditures per legislator does not display this same trend. There does not appear to be a clear geometric division between cluster 1 and 2 for expenditures.

```
# Arizona is the cluster 1 outlier for total session length
which(sub$t_slength > 1 & sub$t_slength < 2) # 3, 35, 38
state_names[3] # Arizona
state_names[35] # Ohio
state_names[38] # Pennsylvania

# Colorado is the cluster 1 outlier for session length
which(sub$slength > .8 & sub$slength < 1.9) #5, 6, 22, 35, 38
state_names[5] # California
state_names[6] # Colorado
state_names[22] # Michigan
state_names[35] # Ohio
state_names[38] # Pennsylvania
```

```r
# Illinois is the cluster 1 outlier for salary
which(sub$salary_real > 1.4 & sub$salary_real < 2.2) #13, 22, 32, 38
state_names[13] # Illinois
state_names[22] # Michigan
state_names[32] # New York
state_names[38] # Pennsylvania
```

Finally, for each of three variables, there is a single state from cluster 1 that appears geometrically located near cluster 2. Arizona is the cluster 1 outlier for total session length, Colorado is the cluster 1 outlier for session length, and Illinois is the cluster 1 outlier for salary.

Question 6

Fit a Gaussian mixture model via the EM algorithm to these data and present the results. Give a quick, high level summary of the output and the general patterns. Initialize the algorithm at k=2, and then check this assumption in the validation questions below.

```r
set.seed(123)
gmm1 <- mvnormalmixEM(data_final,
                      k = 2)
```

```
## number of iterations= 25
```

```r
gmm1$mu
```

```
## [[1]]
## [1] -0.2763465 -0.2450724 -0.1943875 -0.1921007
##
## [[2]]
## [1] 2.431846 2.156634 1.710608 1.690484
```

```r
gmm1$sigma
```

```
## [[1]]
##            [,1]       [,2]      [,3]       [,4]
## [1,] 0.24528126 0.27954563 0.2096713 0.03105197
## [2,] 0.27954563 0.32491503 0.2375517 0.02479181
## [3,] 0.20967129 0.23755167 0.5806866 0.13660923
## [4,] 0.03105197 0.02479181 0.1366092 0.23481667
##
## [[2]]
##           [,1]       [,2]      [,3]       [,4]
## [1,] 0.8556104  1.0197243 0.3979561  0.3509037
## [2,] 1.0197243  1.5611386 0.3426861 -0.3956420
## [3,] 0.3979561  0.3426861 1.2312537  1.9833537
## [4,] 0.3509037 -0.3956420 1.9833537  4.3511252
```

```
gmm1$lambda
```

```
## [1] 0.897959 0.102041
```

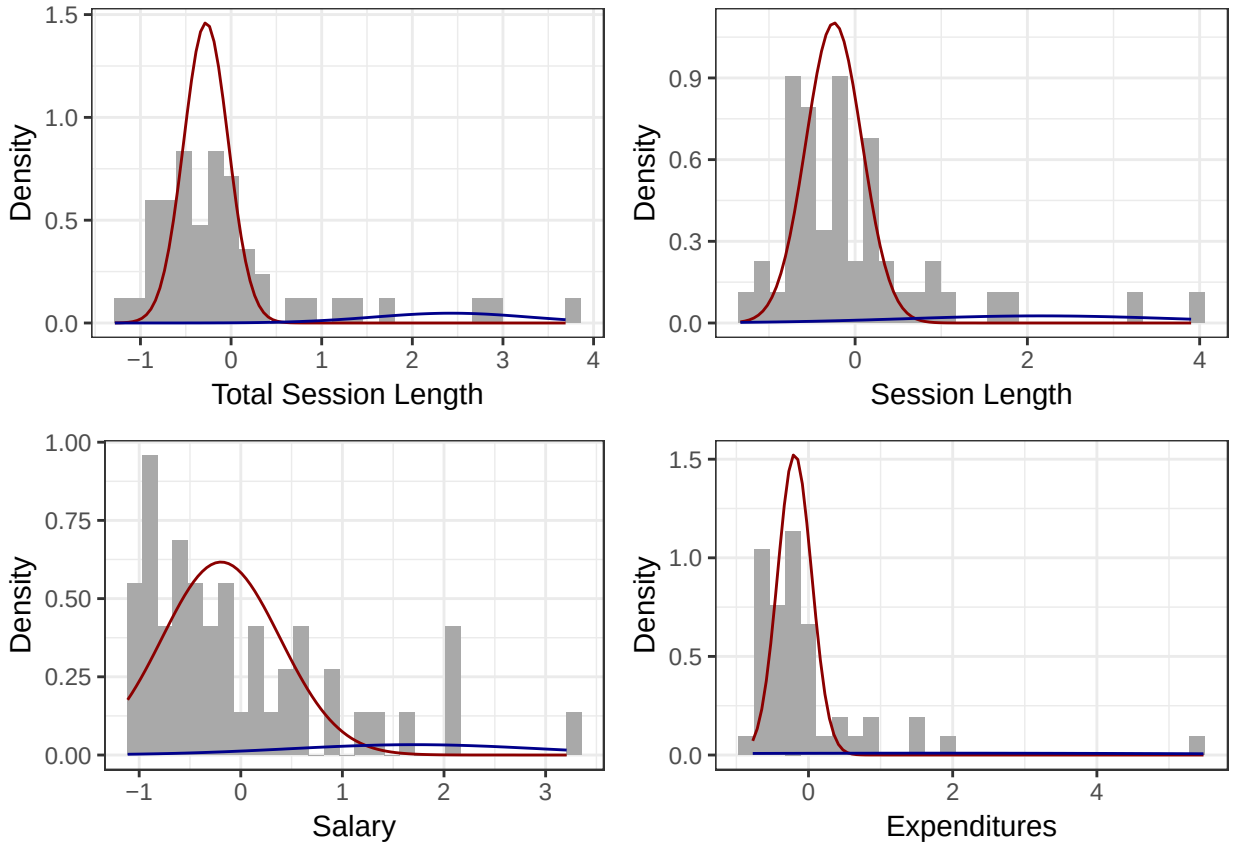
```
posterior <- data.frame(cbind(gmm1$posterior))
rownames(posterior) <- state_names
posterior$component <- ifelse(posterior$comp.1 > 0.3, 1, 2)
data_final_g <- data_final
data_final_g$Cluster <- as.factor(posterior$component)
kable(posterior)
```

	comp.1	comp.2	component
Alabama	1.0000000	0.0000000	1
Alaska	1.0000000	0.0000000	1
Arizona	0.0000000	1.0000000	2
Arkansas	1.0000000	0.0000000	1
California	0.0000000	1.0000000	2
Colorado	1.0000000	0.0000000	1
Connecticut	1.0000000	0.0000000	1
Delaware	0.9999924	0.0000076	1
Florida	1.0000000	0.0000000	1
Georgia	1.0000000	0.0000000	1
Hawaii	1.0000000	0.0000000	1
Idaho	1.0000000	0.0000000	1
Illinois	1.0000000	0.0000000	1
Indiana	1.0000000	0.0000000	1
Iowa	1.0000000	0.0000000	1
Kansas	1.0000000	0.0000000	1
Kentucky	1.0000000	0.0000000	1
Louisiana	1.0000000	0.0000000	1
Maine	1.0000000	0.0000000	1
Maryland	1.0000000	0.0000000	1
Massachusetts	0.0000001	0.9999999	2
Michigan	1.0000000	0.0000000	1
Minnesota	1.0000000	0.0000000	1
Mississippi	1.0000000	0.0000000	1
Missouri	1.0000000	0.0000000	1
Montana	1.0000000	0.0000000	1
Nebraska	1.0000000	0.0000000	1
Nevada	1.0000000	0.0000000	1
New Hampshire	1.0000000	0.0000000	1
New Jersey	1.0000000	0.0000000	1
New Mexico	1.0000000	0.0000000	1
New York	0.0000000	1.0000000	2
North Carolina	1.0000000	0.0000000	1
North Dakota	1.0000000	0.0000000	1
Ohio	1.0000000	0.0000000	1
Oklahoma	1.0000000	0.0000000	1
Oregon	1.0000000	0.0000000	1
Pennsylvania	0.0000007	0.9999993	2
Rhode Island	1.0000000	0.0000000	1
South Carolina	1.0000000	0.0000000	1

	comp.1	comp.2	component
South Dakota	1.0000000	0.0000000	1
Tennessee	1.0000000	0.0000000	1
Texas	1.0000000	0.0000000	1
Utah	1.0000000	0.0000000	1
Vermont	1.0000000	0.0000000	1
Virginia	1.0000000	0.0000000	1
Washington	1.0000000	0.0000000	1
West Virginia	1.0000000	0.0000000	1
Wyoming	1.0000000	0.0000000	1

```
z <- ggplot(data.frame(x = gmm1$x[,1])) +
    geom_histogram(aes(x, ..density..), fill = "darkgray") +
    stat_function(geom = "line", fun = plot_mix_comps,
                  args = list(gmm1$mu[[1]][1], gmm1$sigma[[1]][1,1], lam = gmm1$lambda[1]),
                  colour = "darkred") +
    stat_function(geom = "line", fun = plot_mix_comps,
                  args = list(gmm1$mu[[2]][1], gmm1$sigma[[2]][1,1], lam = gmm1$lambda[2]),
                  colour = "darkblue") +
    xlab("Total Session Length") +
    ylab("Density") +
    theme_bw()

y <- ggplot(data.frame(x = gmm1$x[,2])) +
    geom_histogram(aes(x, ..density..), fill = "darkgray") +
    stat_function(geom = "line", fun = plot_mix_comps,
                  args = list(gmm1$mu[[1]][2], gmm1$sigma[[1]][2,2], lam = gmm1$lambda[1]),
                  colour = "darkred") +
    stat_function(geom = "line", fun = plot_mix_comps,
                  args = list(gmm1$mu[[2]][2], gmm1$sigma[[2]][2,2], lam = gmm1$lambda[2]),
                  colour = "darkblue") +
    xlab("Session Length") +
    ylab("Density") +
    theme_bw()

x <- ggplot(data.frame(x = gmm1$x[,3])) +
    geom_histogram(aes(x, ..density..), fill = "darkgray") +
    stat_function(geom = "line", fun = plot_mix_comps,
                  args = list(gmm1$mu[[1]][3], gmm1$sigma[[1]][3,3], lam = gmm1$lambda[1]),
                  colour = "darkred") +
    stat_function(geom = "line", fun = plot_mix_comps,
                  args = list(gmm1$mu[[2]][3], gmm1$sigma[[2]][3,3], lam = gmm1$lambda[2]),
                  colour = "darkblue") +
    xlab("Salary") +
    ylab("Density") +
    theme_bw()

w <- ggplot(data.frame(x = gmm1$x[,4])) +
    geom_histogram(aes(x, ..density..), fill = "darkgray") +
    stat_function(geom = "line", fun = plot_mix_comps,
                  args = list(gmm1$mu[[1]][4], gmm1$sigma[[1]][4,4], lam = gmm1$lambda[1]),
                  colour = "darkred") +
```

```
    stat_function(geom = "line", fun = plot_mix_comps,
                  args = list(gmm1$mu[[2]][4], gmm1$sigma[[2]][4,4], lam = gmm1$lambda[2]),
                  colour = "darkblue") +
    xlab("Expenditures") +
    ylab("Density") +
    theme_bw()

grid.arrange(z, y, x, w, ncol=2, nrow=2)
```



Fitting a Guassian mixture model via the EM algorithm produces interesting results, as we are assuming a mixture of normally distributed clusters, which may not be a valid assumption in this case. Observations have been given probabilities of belonging to all clusters and then were clustered on the basis of probabilistic similarities. It took 25 iterations to establish stable cluster assignments (convergence). Two components emerged, one with means just below the average (falling between -0.28 and -0.19), and one with means far greater than the average (falling between 1.7 and 2.4). These estimates coupled with the plots above demonstrate that although one cluster may be normally distributed, the second does not appear to be normally distributed due to several high-value outliers. As a result, the distribution with higher means has a very high spread. The initial value of mixing proportions or density encapsulation was 89.8% and 10.2%. Each data was assigned a posterior probability of belonging to one of the components, which is summarized in the table above. It is interesting to note that removing outliers may have produced bi-modal distributions more favorable to using a Guassian mixure model. Using a threshold of 0.3, 44 states were assigned to one component and 5 states were assigned to the second (California, Massachusetts, New York, Pennsylvania, and Arizona).

Question 7

Fit one additional partitioning technique of your choice (eg. PAM, CLARA, fuzzy C-means, DBSCAN, etc.) and present and discuss results. Here again initialize at k=2.

```
# Clara (Clustering Large)
set.seed(2)
data_final <- sub %>% select(-state)
clara <- clara(data_final, 2,
               samples = 15)
str(clara)
```

```
## List of 10
##  $ sample    : int [1:44] 2 3 4 5 6 7 8 9 11 12 ...
##  $ medoids   : num [1:2, 1:4] -0.295 0.776 -0.21 1.006 -0.579 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:4] "t_slength" "slength" "salary_real" "expend"
##  $ i.med     : int [1:2] 39 22
##  $ clustering: int [1:49] 1 1 1 1 2 1 1 1 1 1 ...
##  $ objective : num 1.17
##  $ clusinfo  : num [1:2, 1:4] 42 7 2.239 5.602 0.956 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:4] "size" "max_diss" "av_diss" "isolation"
##  $ diss      : 'dissimilarity' num [1:946] 2.38 1.67 6.53 1.86 1.2 ...
##   ..- attr(*, "Size")= int 44
##   ..- attr(*, "Metric")= chr "euclidean"
##  $ call      : language clara(x = data_final, k = 2, samples = 15)
##  $ silinfo   :List of 3
##   ..$ widths         : num [1:44, 1:3] 1 1 1 1 1 1 1 1 1 1 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:44] "39" "42" "48" "17" ...
##   .. .. ..$ : chr [1:3] "cluster" "neighbor" "sil_width"
##   ..$ clus.avg.widths: num [1:2] 0.702 0.135
##   ..$ avg.width      : num 0.612
##  $ data      : num [1:49, 1:4] -0.372 -0.229 1.645 -0.804 2.881 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:4] "t_slength" "slength" "salary_real" "expend"
##  - attr(*, "class")= chr [1:2] "clara" "partition"
```

```
kable(clara$medoids)
```

t_slength	slength	salary_real	expend
-0.2949443	-0.2101066	-0.5789619	-0.3295059
0.7755062	1.0063116	2.1381147	0.4568995

```
kable(clara$clusinfo)
```

size	max_diss	av_diss	isolation
42	2.239065	0.9555198	0.6868659
7	5.602221	2.4390529	1.7185637

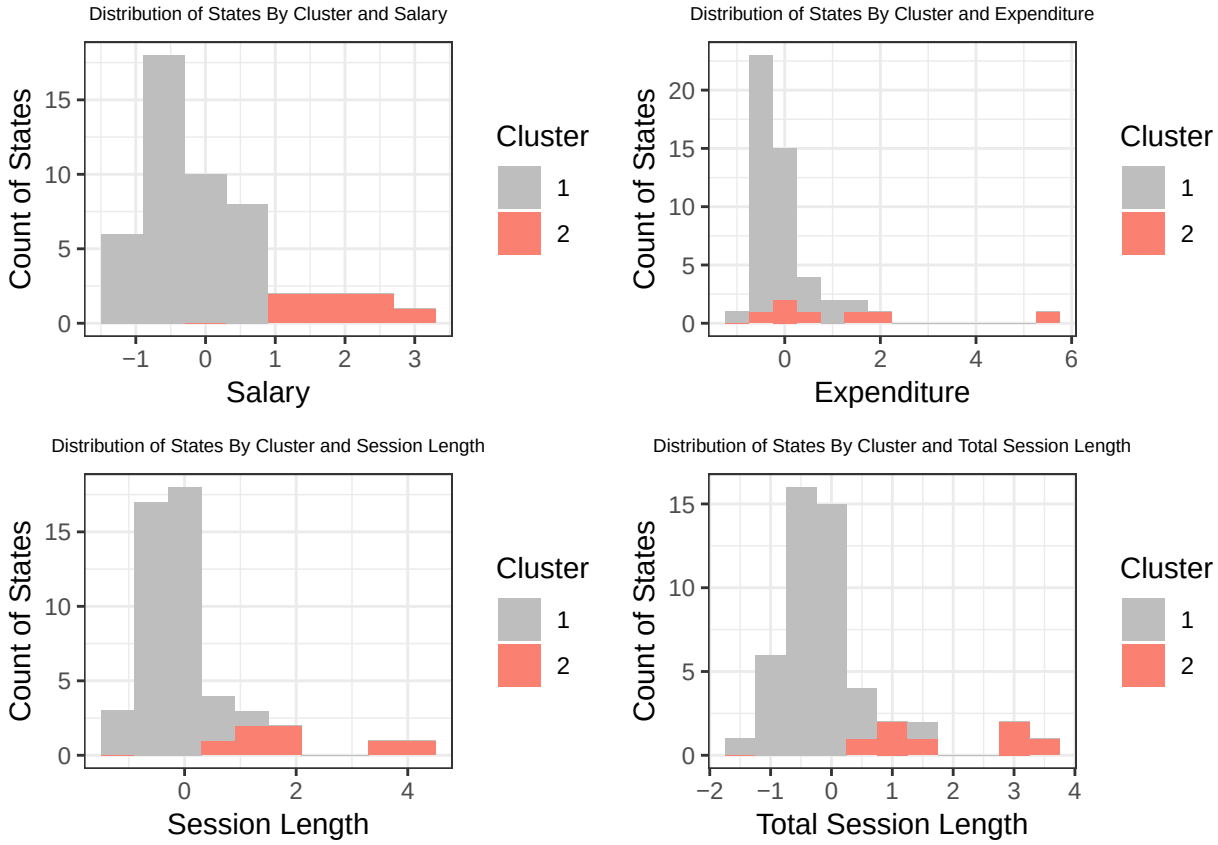
```
data_final_c <- data_final
data_final_c$Cluster <- as.factor(clara$cluster)
t <- as.table(clara$cluster)
t <- data.frame(t)
rownames(t) <- state_names
colnames(t)[colnames(t) == "Freq"] <- "Assignment"
t$Var1 <- NULL

kable(t)
```

	Assignment
Alabama	1
Alaska	1
Arizona	1
Arkansas	1
California	2
Colorado	1
Connecticut	1
Delaware	1
Florida	1
Georgia	1
Hawaii	1
Idaho	1
Illinois	2
Indiana	1
Iowa	1
Kansas	1
Kentucky	1
Louisiana	1
Maine	1
Maryland	1
Massachusetts	2
Michigan	2
Minnesota	1
Mississippi	1
Missouri	1
Montana	1
Nebraska	1
Nevada	1
New Hampshire	1
New Jersey	1
New Mexico	1
New York	2
North Carolina	1
North Dakota	1

	Assignment
Ohio	2
Oklahoma	1
Oregon	1
Pennsylvania	2
Rhode Island	1
South Carolina	1
South Dakota	1
Tennessee	1
Texas	1
Utah	1
Vermont	1
Virginia	1
Washington	1
West Virginia	1
Wyoming	1

```
# Distribution of states based on their cluster assignment
e <- ggplot(data_final_c, aes(salary_real, fill = Cluster)) +
    geom_histogram(binwidth = 0.6) +
    theme_bw() +
    scale_fill_manual(values=c("grey", "salmon")) +
    labs(x = "Salary",
         y = "Count of States",
         title = "Distribution of States By Cluster and Salary") +
    theme(plot.title = element_text(hjust = 0.5)) +
    theme(plot.title = element_text(size=7))

f <- ggplot(data_final_c, aes(expend, fill = Cluster)) +
    geom_histogram(binwidth = 0.5) +
    theme_bw() +
    scale_fill_manual(values=c("grey", "salmon")) +
    labs(x = "Expenditure",
         y = "Count of States",
         title = "Distribution of States By Cluster and Expenditure") +
    theme(plot.title = element_text(hjust = 0.5)) +
    theme(plot.title = element_text(size=7))

g <- ggplot(data_final_c, aes(slength, fill = Cluster)) +
    geom_histogram(binwidth = 0.6) +
    theme_bw() +
    scale_fill_manual(values=c("grey", "salmon")) +
    labs(x = "Session Length",
         y = "Count of States",
         title = "Distribution of States By Cluster and Session Length") +
    theme(plot.title = element_text(hjust = 0.5)) +
    theme(plot.title = element_text(size=7))

h <- ggplot(data_final_c, aes(t_slength, fill = Cluster)) +
    geom_histogram(binwidth = 0.5) +
    theme_bw() +
    scale_fill_manual(values=c("grey", "salmon")) +
    labs(x = "Total Session Length",
```

```
        y = "Count of States",
        title = "Distribution of States By Cluster and Total Session Length") +
    theme(plot.title = element_text(hjust = 0.5)) +
    theme(plot.title = element_text(size=7))

grid.arrange(e, f, g, h, ncol=2, nrow=2)
```



Partitioning around medoids (PAM) is similar to k-means, and is a hard partitioning algorithm that aims to minimize within cluster variation. In k-medoids, a medoid is a data point from a dataset whose average dissimilarity to all other data points is minimal. Althought Clara is an extension of PAM intended for large datasets, I was curious to see its output on a smaller dataset as an illustrative example. In this case, the data have been separated into two clusters, with 7 states belonging to cluster 2 (California, Massachusetts, Michigan, New York, Ohio, Pennsylvania, and Illinois), and the the rest of the states belonging to cluster 1. For cluster 2 (with 7 states), the medoids are high values (0.78 total session length, 1.01 session length, 2.14 salary, and 0.46 expenditures). For cluster 1 (with 42 states), the medoids are lower values close to the mean (-0.29 for total session length, -0.21 for session length, -0.58 for salary, and -0.33 for expenditures). Furthermore, for cluster 2, the maximum pairwise dissimilarity is 5.6, wereas for cluster 1, the maximum pairwise dissimilarity is 2.2, indicating a potentially more compact cluster. By plotting the distribution of the four variables of interest based on cluster assignment, patterns emerge. For salary, all of the states in cluster 2 appear to the left of cluster 1 (higher values). However, this is not the case for the other three variables, which display overlap between the two clusters. In the expenditure distribution, there is no noticable geometric division between the two clusters.

Question 8

Compare output of all in a visually useful, simple way (eg. plotting by state cluster assignment across two features like salary and expenditures).

Note: See visualizations included in Questions 5, 6, and 7 as well.

```
j <- ggplot(data_final_k, aes(x = salary_real, y = expend, col = Cluster)) +
  geom_point() +
  labs(x = "Salary",
       y = "Expenditures",
       title = "K-Means: Salary & Expend") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(plot.title = element_text(size=10))

k <- ggplot(data_final_g, aes(x = salary_real, y = expend, col = Cluster)) +
  geom_point() +
  labs(x = "Salary",
       y = "Expenditures",
       title = "Gaussian: Salary & Expend") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(plot.title = element_text(size=10))

l <- ggplot(data_final_c, aes(x = salary_real, y = expend, col = Cluster)) +
  geom_point() +
  labs(x = "Salary",
       y = "Expenditures",
       title = "Clara: Salary & Expend") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(plot.title = element_text(size=10))

m <- ggplot(data_final_k, aes(x = t_slength, y = slength, col = Cluster)) +
  geom_point() +
  labs(x = "Salary",
       y = "Expenditures",
       title = "K-Means: Total & Sess. Length") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(plot.title = element_text(size=10))

n <- ggplot(data_final_g, aes(x = t_slength, y = slength, col = Cluster)) +
  geom_point() +
  labs(x = "Salary",
       y = "Expenditures",
       title = "Gaussian: Total & Sess. Length") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(plot.title = element_text(size=10))

o <- ggplot(data_final_c, aes(x = t_slength, y = slength, col = Cluster)) +
```

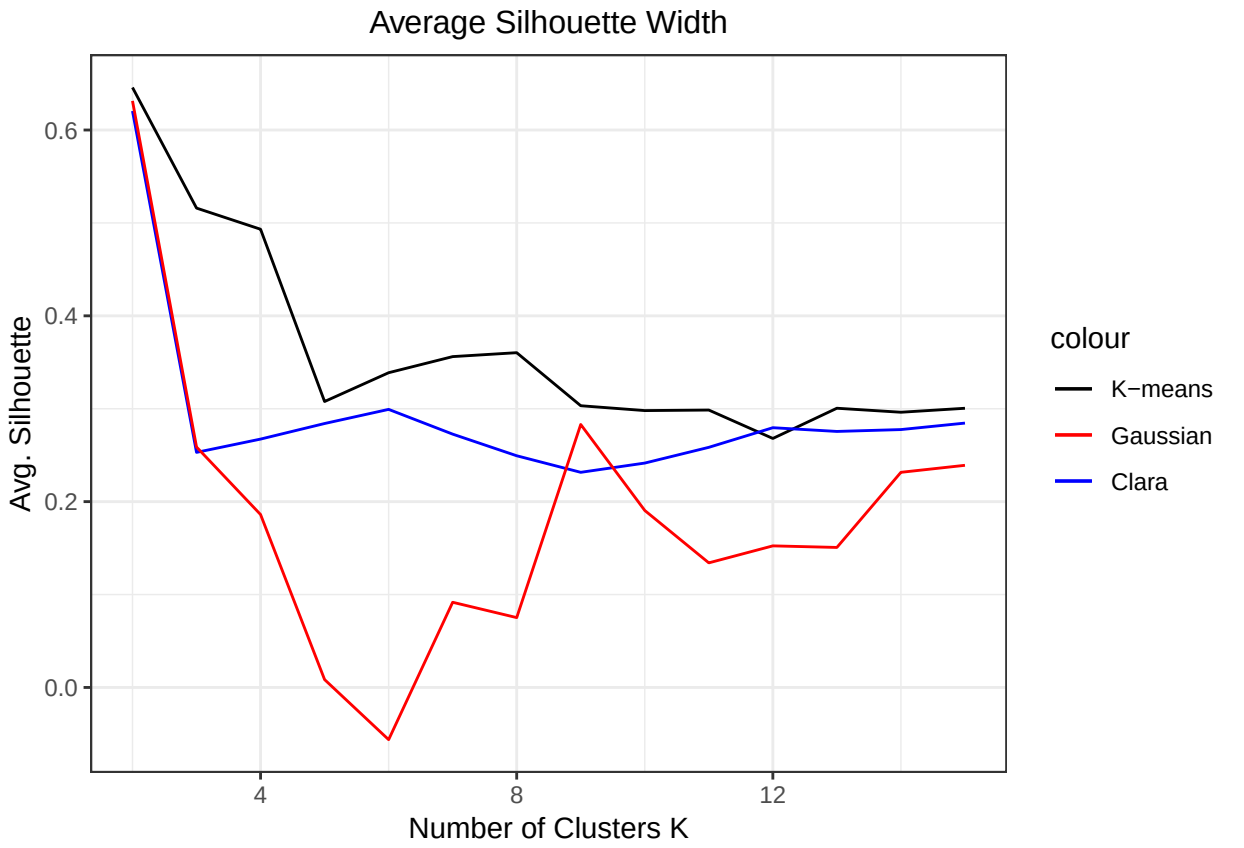
```
  geom_point() +
  labs(x = "Salary",
       y = "Expenditures",
       title = "Clara: Total & Sess. Length") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(plot.title = element_text(size=10))

grid.arrange(j, k, l, m, n, o, ncol=3, nrow=2)
```

Plotting by state cluster assignment across pairs of two features (eg. salary and expenditures; total session length and session length) delivers two key findings. First, for these pairs of variables, one cluster is always located in the bottom right of the visualization and appears dense, while the second cluster is always located in the upper left of the visualization and appears sparse. Second, comparing the three clustering algorithms, there are notable discrepancies among states at the margin.

Question 9

Select a single validation strategy (eg. compactness via min(WSS), average silhouette width, etc.), and calculate for all three algorithms. Display and compare your results for all three algorithms you fit (k-means, GMM, and Clara).

```r
set.seed(123)
# Function to compute avg. silhouette width (kmeans)
avg_sil <- function(k) {
           km.res <- kmeans(data_final, k, nstart=15)
           s <- silhouette(km.res$cluster, dist(data_final))
           mean(s[,3])
}

# Function to compute avg. silhouette width (Clara)
avg_sil2 <- function(k) {
           clara <- clara(data_final, k)
           s <- silhouette(clara$cluster, dist(data_final))
           mean(s[,3])
}

# Function to compute avg. silhouette width (Gaussian)
avg_sil3 <- function(k) {
           gmm <- Mclust(data_final, G=k)
           cs <- cluster.stats(dist(data_final), gmm$classification)
           cs$avg.silwidth
}

# Compute for k = 2 to k = 15
k.values <- 2:15
s_values <- map_dbl(k.values, avg_sil)
df <- as.data.frame(cbind(s_values, k.values))
s_values <- map_dbl(k.values, avg_sil2)
df2 <- as.data.frame(cbind(s_values, k.values))
s_values <- map_dbl(k.values, avg_sil3)
df3 <- as.data.frame(cbind(s_values, k.values))

ggplot(df, aes(x=k.values, y=s_values, colour = "black")) +
      geom_line(data = df, aes()) +
      theme_bw() +
      ggtitle("Average Silhouette Width") +
      xlab("Number of Clusters K") +
      ylab("Avg. Silhouette") +
      theme(plot.title = element_text(hjust = 0.5)) +
      theme(plot.title = element_text(size=12)) +
      geom_line(data = df2, aes(x=k.values, y=s_values, colour = "red")) +
      geom_line(data = df3, aes(x=k.values, y=s_values, colour = "blue")) +
      scale_color_manual(values = c('black', 'red', "blue"),
                         labels = c("K-means", "Gaussian", "Clara"))
```



One validation strategy is computing the average silhouette width, which describes how well each observation lies within its cluster across many values of k. In other words, it tells us how well-defined clusters are, where a high average silhouette width indicates a good confguration. Across all three algorithms, k=2 consistently has he highest average silhouette width, indicating that it is likely the best cluster configuration relative to other values of k (validating our prior assumptions). Furthermore, of these algorithms, K-means consistently has the highest average silhouette values (for k=2 through k=15), indicating that this algorithm performs the best relative to the other algorithms. Meanwhile, Gaussian has inconsistent performance at different levels of k, while Clara has the most consistent performance and is likely the second best cluster configuration after it diverges from Gaussian at k=3. By k=15, all three algorithms begin to converge to the same average silhouette width.

Question 10

Discuss the validation output.

a. What can you take away from the fit?

Based on this comparison, it is unlikely that the data represent a mixture of normally distributed clusters (or Gaussians), given that GMM consistently has the lowest average silhouette width over most values of k (except for k=9). Furthermore, it is likely that there are 2 natural, non-random clusters in the data, since k=2 has the highest average silhouette width across all three models. Based on this validation process, k-means appears to be the optimal clustering algorithm for this data. For k-means at k=2, the average silhouette width is just over 0.6, and observations with a silhouette width close 1 are considered very well clustered. Therefore, these data are relatively well clustered at k=2.

b. Which approach is optimal? And optimal at what value of k?

Based on this comparison, I conclude that that k-means is the optimal clustering algorithm for this data (among k-means, Gaussian, and clara) at k=2.

c. What are reasons you could imagine selecting a technically "sub-optimal" partitioning method, regardless of the validation statistics?

There are several reasons one might select a technically "sub-optimal" partitioning method. Internal validation compares across different algorithms and reports the "best one" dependent on the selected metric. Selecting a metric is subjective, and it is possible that a "sub-optimal" partitioning method for one metric may have performed well for metrics besides those tested. Moreover, one might select a "sub-optimal" method due to computational resource constraints or based on performance requirements. In other words, the method with the best validation statistics may actually be computationally expensive and/or take a much longer time to run on the entire dataset, if only tested with a sample. Finally, applying unsupervised methods requires a level of domain expertise. It is possible that a domain expert may emphasize that observations cannot belong to more than one cluster, for instance, and therefore may shy away from using an algorithm like GMM even if it has the best validation statistics.