Assignment3_AbhishekPandit

Abhishek Pandit

25 October 2019

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.

We will be using mostly base R with a few specialized libraries.

Now we load the data:

```
legprof<- read.csv('leg_prof.csv') #loaded from RWorkspace using wroite.csv
set.seed(1984)
```

*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)*

First, we explore some of the variables

```
unique(legprof$year)
```

```
##  [1] 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000
## [15] 2002 2004 2006 2008 2010 1975 1977 1979 1981 1983 1985 1987 1989 1991
## [29] 1993 1995 1997 1999 2001 2003 2005 2007 2009 2011
```

```
unique(legprof$sessid)
```

```
##  [1] 1973/4  1975/6  1977/8  1979/80 1981/2  1983/4  1985/6  1987/7
##  [9] 1989/90 1991/2  1993/4  1995/6  1997/8  1999/00 2001/2  2003/4
## [17] 2005/6  2007/8  2009/10
## 19 Levels: 1973/4 1975/6 1977/8 1979/80 1981/2 1983/4 1985/6 ... 2009/10
```

As expected from the logbook, we see years at intervals of 2. In this sense, it will be wiser to use the variable 'sessid' for us to focus on one year.

Now, onto the munging. We will need states later for plotting purposes. So it would make sense to save the column separately for now and then reunite it with the main data frame.

```
legprof.all <- legprof %>%
  filter(sessid=="2009/10") %>%
  dplyr::select(expend, t_slength, slength, salary_real, stateabv)%>%
  na.omit() %>%
  mutate_if(is.numeric, scale)%>%
  as.data.frame()
```

```
## Warning: package 'bindrcpp' was built under R version 3.5.2
```

```
legprof.state<- legprof.all[,"stateabv"]
```

```
legprof_sub <-legprof.all %>%
  dplyr::select(expend, t_slength, slength, salary_real)%>%
  as.data.frame()

head(legprof.all)
```

```
##       expend  t_slength    slength salary_real stateabv
## 1 -0.2399910 -0.3716599 -0.4594723  -1.0920009       AL
## 2  0.8591198 -0.2294089 -0.1452309   0.4011333       AK
## 3 -0.1299408  1.6453067  0.7951955  -0.1335656       AZ
## 4 -0.2612061 -0.8036462 -0.7881756  -0.4923902       AR
## 5  5.4785453  2.8807257  1.7767099   3.2069914       CA
## 6 -0.3485530  0.6827338  0.9008887   0.1113595       CO
```

We check if all 50 states are covered.

```
dim(legprof_sub)
```

```
## [1] 49  4
```

This shows that one state is either missing or was dropped due to missing data. A quick glance the data reveals that this has indeed been the case for Wisconsin. So we will proceed with our 49 states.

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

```
summary(legprof_sub)
```

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

Based on the quartile distribution, salary and expenditure- the economic variables seem highly skewed. The variables on time, conversely, seem closer to normally distributed.

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

First, we do so informally by observing pairwise scatterplots.

```
library(skimr, quietly=TRUE)
```

```
## Warning: package 'skimr' was built under R version 3.5.3
```

```
##
## Attaching package: 'skimr'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
skim(as.data.frame(legprof_sub))
```

```
## Warning: No summary functions for vectors of class: matrix.
## Coercing to character
```

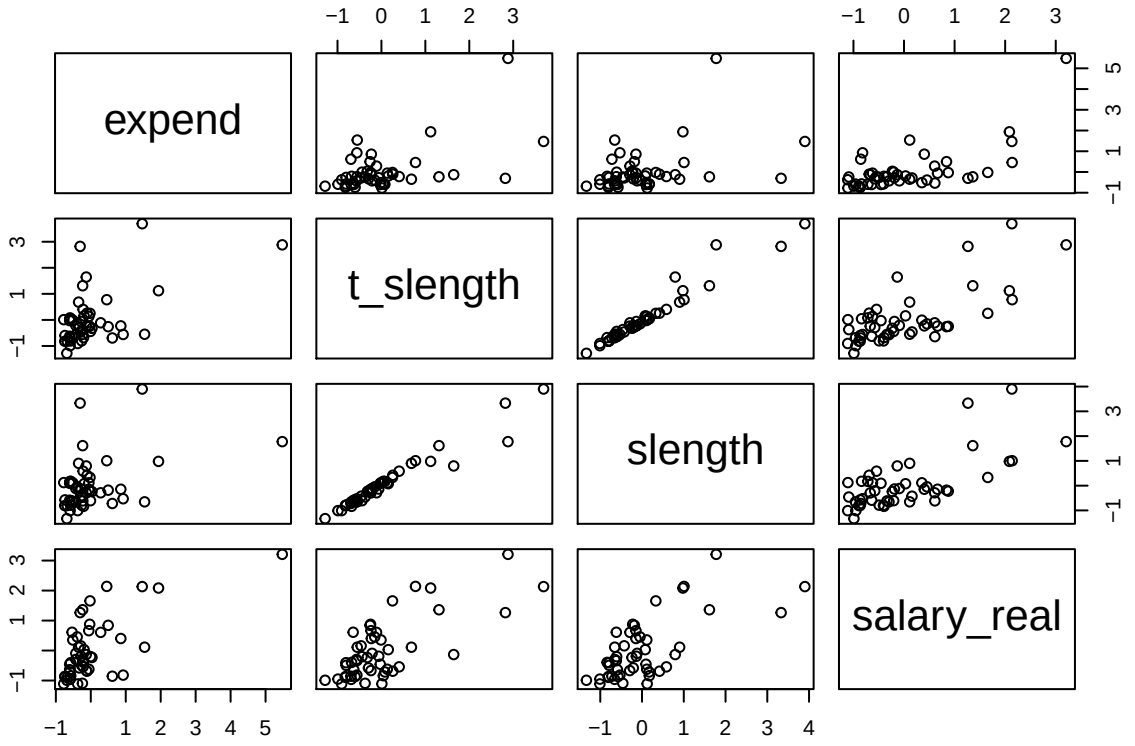
```
## Warning: No summary functions for vectors of class: matrix.
## Coercing to character
```

```
## Warning: No summary functions for vectors of class: matrix.
## Coercing to character
```

```
## Warning: No summary functions for vectors of class: matrix.
## Coercing to character
```

```
## Skim summary statistics
##  n obs: 49
##  n variables: 4
##
## -- Variable type:character ------------------------------------------------------------------------
##     variable missing complete  n min max empty n_unique
##       expend       0       49 49  16  20     0       49
##  salary_real       0       49 49  15  19     0       49
##      slength       0       49 49  16  19     0       44
##    t_slength       0       49 49  16  20     0       48
```

```
pairs(legprof_sub)
```



Interestingy, there to be some positive relationship between salary and the two types of session lengths.

Expenditure seems to be related positively with all three of the other variables, but with a small slope.

In almost all the plots, there is a set of 6-8 outliers towards the upper right of the origin. These outliers do not seem close enough to themselves form a cluster. However, the remaining points do form one close cluster in each plot. Whether this relationship holds in higher dimensions remains to be seen.

Not surprisingly, there is a strong correlation between session length and total session length Let's check this numerically.

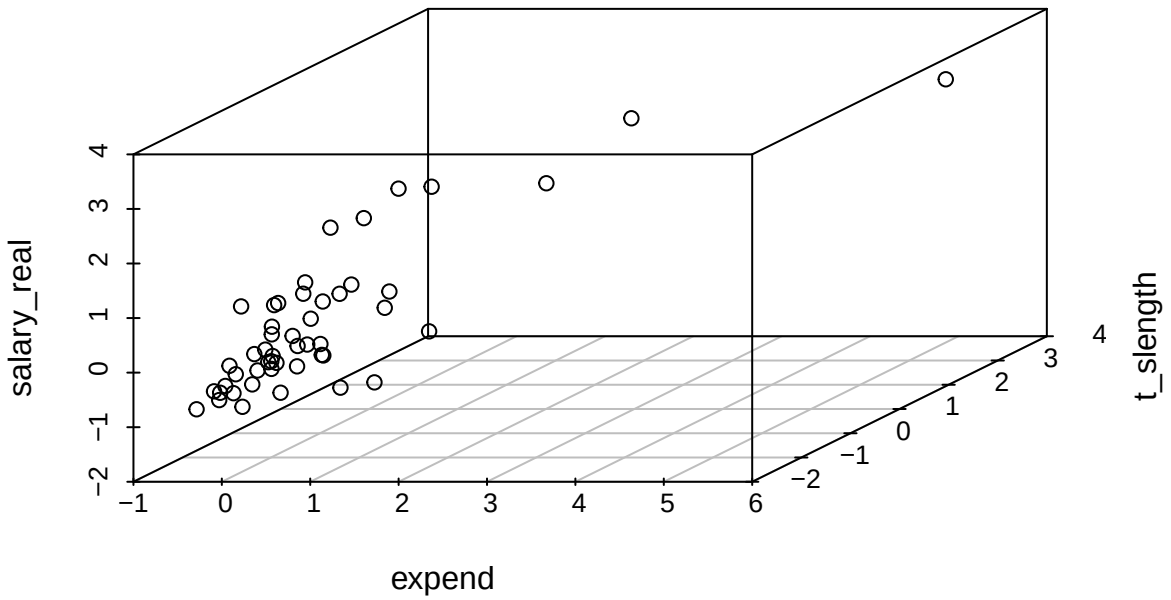
```
cor(legprof_sub$t_slength, legprof_sub$slength)
```

```
##           [,1]
## [1,] 0.9708659
```

97% correlation suggests a perfectly linear and positive relationship. In this sense, one of the variables can be easily dropped. We will return to this point in later analyses.

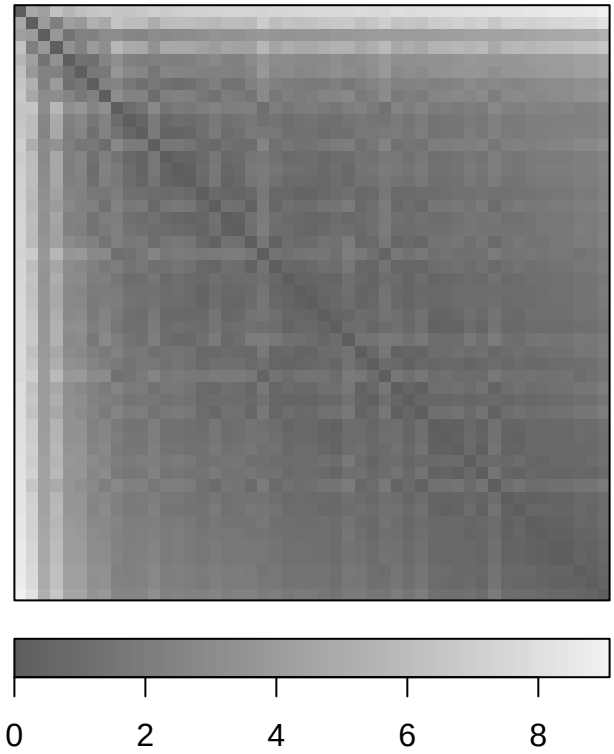
If we drop one of them, we will be to check for 3-d scatter plots. So for the sake of experimentation, let's try a 3d version

```
legprof_3d<-legprof_sub %>%
  dplyr::select(expend, t_slength, salary_real)
scatterplot3d(legprof_3d)
```



Just lke in the 2d case, we see a strong clustering in one corner of the plot, with about 8 outliers.

Now we turn to more formal methods such as the ODI.



Again, the majority of the data seems to form one cluster, with one less dense cluster in the upper left corner.

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.

```
leg_kmeans <- kmeans(legprof_sub,
                     centers = 2,
                     nstart = 15)
leg_kmeans$size
```

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

```
kmeans_table <-as.data.frame(cbind(legprof.all$stateabv,leg_kmeans$cluster))
names(kmeans_table) <- c("State", "Cluster")
```

As per this first classification, we have two clusters- with the majority in the first (43 states) and a minority in the second one.

```
outliers<-kmeans_table %>%
  filter(Cluster==2)
outliers
```

```
##   State Cluster
## 1     5       2
## 2    19       2
## 3    22       2
## 4    34       2
## 5    35       2
```

```
## 6    38       2
```

So as per this classification, the second distinct cluster of states in terms of legislative professionalism is California, Maine, Minnesota, New York, Ohio and Pennsylvania. There does not seem to be a strong geographic pattern to them.

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 general patterns. Initialize the algorithm at k=2, and then check this assumption in the validation questions below

```
set.seed(123)
library(mixtools)
gmm_leg<-mvnormalmixEM(as.matrix(legprof_sub),lambda = NULL, mu = NULL, sigma = NULL, k = 2)
```

```
## number of iterations= 16
```

```
gmm_leg$posterior
```

```
##              comp.1      comp.2
##  [1,]  9.960442e-01 0.003955767
##  [2,]  3.778750e-09 0.999999996
##  [3,]  8.307371e-68 1.000000000
##  [4,]  9.920070e-01 0.007993044
##  [5,] 3.628290e-127 1.000000000
##  [6,]  9.621328e-01 0.037867167
##  [7,]  9.698649e-01 0.030135118
##  [8,]  9.960809e-01 0.003919105
##  [9,]  5.331047e-21 1.000000000
## [10,]  9.982789e-01 0.001721145
## [11,]  9.958085e-01 0.004191517
## [12,]  9.966146e-01 0.003385411
## [13,]  9.795616e-01 0.020438417
## [14,]  9.917046e-01 0.008295439
## [15,]  9.981832e-01 0.001816798
## [16,]  9.968360e-01 0.003163968
## [17,]  9.965070e-01 0.003492964
## [18,]  9.901724e-01 0.009827553
## [19,]  9.985438e-01 0.001456172
## [20,]  9.881357e-01 0.011864312
## [21,]  1.517471e-10 1.000000000
## [22,]  2.367535e-02 0.976324648
## [23,]  9.983060e-01 0.001693994
## [24,]  9.955047e-01 0.004495307
## [25,]  9.945283e-01 0.005471722
## [26,]  9.984606e-01 0.001539423
## [27,]  9.913903e-01 0.008609682
## [28,]  2.764356e-07 0.999999724
## [29,]  9.913494e-01 0.008650620
## [30,]  5.773349e-03 0.994226651
## [31,]  9.970746e-01 0.002925365
## [32,]  9.116708e-20 1.000000000
## [33,]  9.839411e-01 0.016058905
## [34,]  9.981656e-01 0.001834379
## [35,]  6.189774e-02 0.938102255
## [36,]  9.970931e-01 0.002906942
## [37,]  8.975502e-01 0.102449828
```

```
## [38,]  6.300173e-16 1.000000000
## [39,]  9.974173e-01 0.002582656
## [40,]  9.626155e-01 0.037384518
## [41,]  9.976825e-01 0.002317529
## [42,]  9.984228e-01 0.001577176
## [43,]  4.578871e-14 1.000000000
## [44,]  9.977183e-01 0.002281688
## [45,]  9.976995e-01 0.002300508
## [46,]  9.960782e-01 0.003921777
## [47,]  9.238639e-01 0.076136129
## [48,]  9.977785e-01 0.002221475
## [49,]  9.946706e-01 0.005329421
```

The small number of iterations (16) suggests that convergence occurred early for this model. Thus, k= 2 may seem justified.

We can also take a look at the probabilities of assignment to one group or the other above. For the majority of states, the mixing probability for one of the two clusters is very high, while the othe is usually very small. This suggests that we are dealing with a case more likely to benefit from hard rather than soft partitioning. We can use this to derive the clustering allocations.

```
posterior <- as.data.frame(cbind(legprof.all$stateabvx, gmm_leg$posterior))
posterior$component <- ifelse(posterior$comp.1 > 0.3, 1, 2)
posterior <-as.data.frame(cbind(as.array(legprof.all$stateabv), posterior))
names(posterior) <-c('State', 'Component1', 'Component2', 'Component')
gmm_leg_cluster <-posterior$Component
gmm_outliers <-posterior %>%
  dplyr::select(State, Component) %>%
  filter(Component==2)
gmm_outliers
```

```
##    State Component
## 1     AK         2
## 2     AZ         2
## 3     CA         2
## 4     FL         2
## 5     MA         2
## 6     MI         2
## 7     NV         2
## 8     NJ         2
## 9     NY         2
## 10    OH         2
## 11    PA         2
## 12    TX         2
```

We find some states to match from the old classification, but there are also new entries like Florida, Nevada, New Jersey, Arkansas and Arizona!

7. Fit one additional partitioning technique of your choice (e.g., PAM, CLARA, fuzzy Cmeans, DBSCAN, etc.), and present and discuss results. Here again initialize at k=2

We now experiment with another unsupervised method, PAM. We can now check which states were found to belong to the second cluster.

```
library(cluster)
```

```
## Warning: package 'cluster' was built under R version 3.5.3
```

```
pam_legprof <- pam(legprof_sub, 2)
pam_legprof
```

```
## Medoids:
##      ID     expend  t_slength    slength salary_real
## [1,] 39 -0.3295059 -0.2949443 -0.2101066  -0.5789619
## [2,] 22  0.4568995  0.7755062  1.0063116   2.1381147
## Clustering vector:
##  [1] 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 1 1 2
## [36] 1 1 2 1 1 1 1 1 1 1 1 1 1 1
## Objective function:
##    build     swap
## 1.188069 1.167453
##
## Available components:
##  [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
##  [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```

```
pam_leg_outliers <- as.data.frame(cbind(legprof.all$stateabv, pam_legprof$clustering))
names(pam_leg_outliers) <-c('State', 'Cluster')
pam_leg_outliers <-pam_leg_outliers %>%
  filter(Cluster==2)
pam_leg_outliers
```

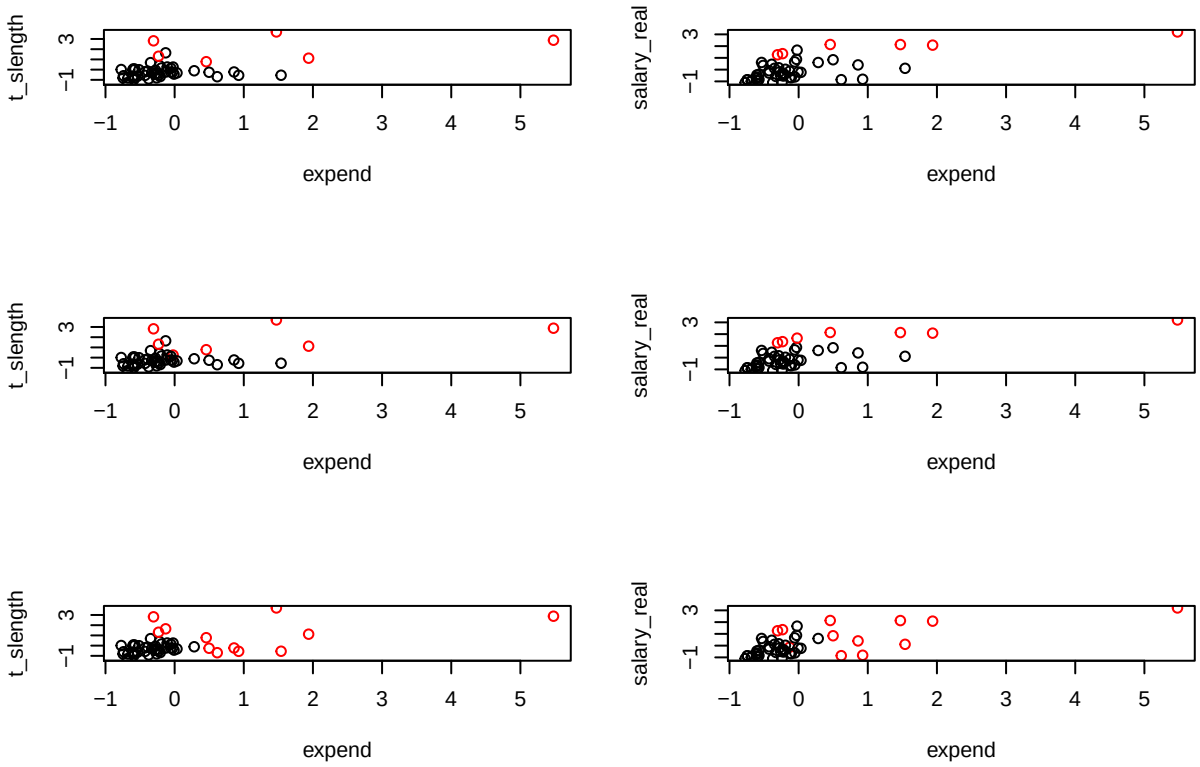
```
##   State Cluster
## 1     5       2
## 2    14       2
## 3    19       2
## 4    22       2
## 5    34       2
## 6    35       2
## 7    38       2
```

The members of the 2nsd cluster here are very similar to those for K-Means. This suggests that the 'means' chosen within each cluster for k-means correspond closely with actual data points (states) chosen as medoids.

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

We will compare for 2 scatterplots of interest across all three methods- specifically, expenditure with total session length.

```
par(mfrow=c(3,2))
plot(legprof_sub[c(1,2)], col=leg_kmeans$cluster)
plot(legprof_sub[c(1,4)], col=leg_kmeans$cluster)
plot(legprof_sub[c(1,2)], col=pam_legprof$clustering)
plot(legprof_sub[c(1,4)], col=pam_legprof$clustering)
plot(legprof_sub[c(1,2)], col=gmm_leg_cluster)
plot(legprof_sub[c(1,4)], col=gmm_leg_cluster)
```



We can see that the classification with two clusters is roughly the same across the first and second row of plots (PAM and K Means) respectively. GMM, however, chooses to include many of the points lower along the y axis that were ignored by the previous 2 algorithms.

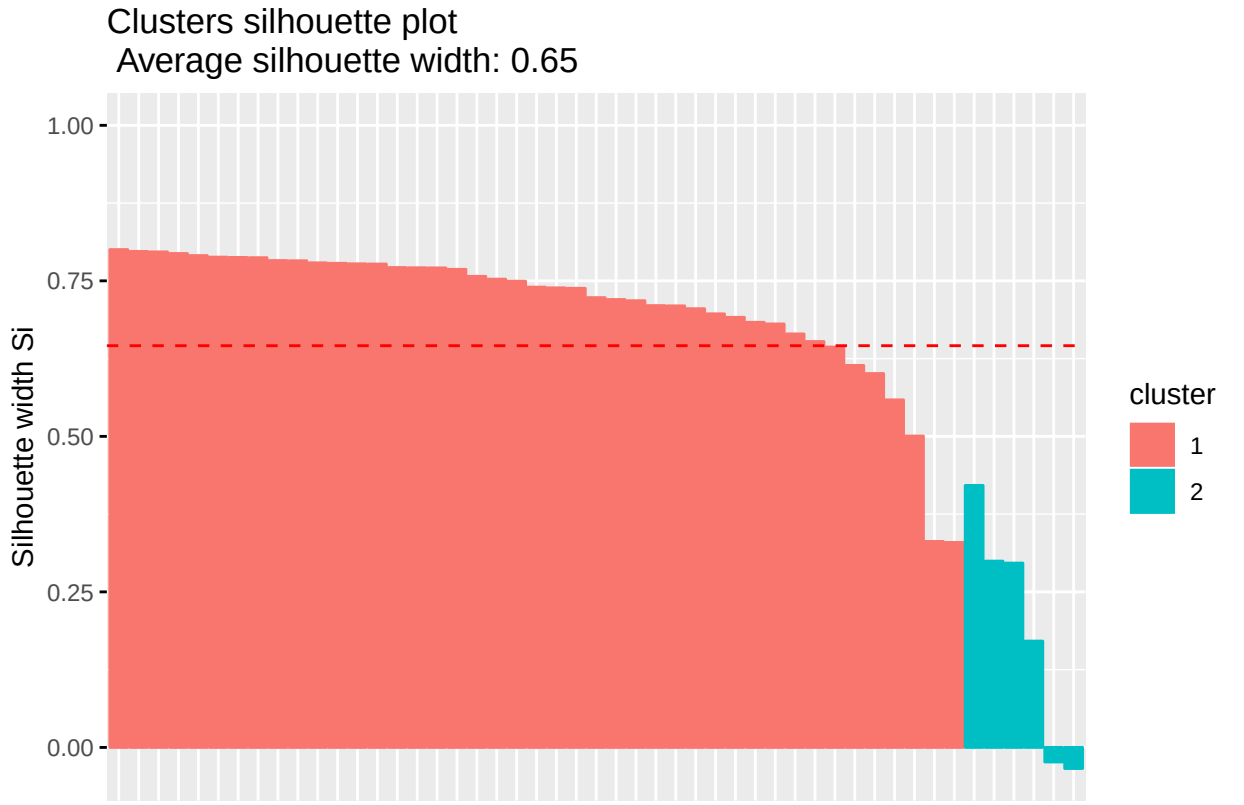
9. Select a single validation strategy (e.g., compactness via min(WSS), average silhouette width, etc.), and calculate for all three algorithms. Display and compare your results for all three algorithms you fit (k-means, GMM, X).

We will be comparing for three strategies across the method of average silhouette width.

First for KMeans:

```
km.sil<-silhouette(leg_kmeans$cluster,dist=legprof_dist )
fviz_silhouette(km.sil)
```

```
##   cluster size ave.sil.width
## 1       1   43          0.71
## 2       2    6          0.19
```

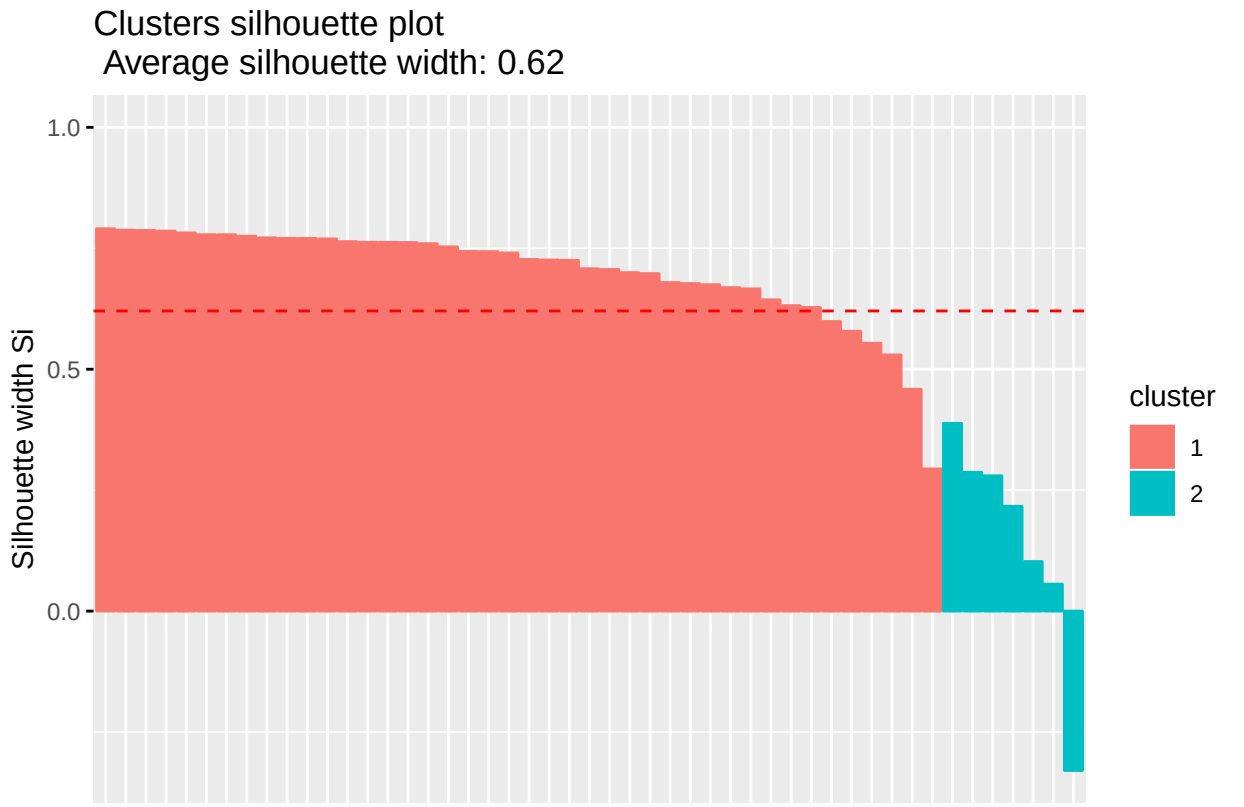


The first cluster, as expected, has a strong score (close to 1). However, the score for the second cluster is weak (near 0)- and suggests that this cluster may lie on the boundary between 2 clusters. Furthermore, two states have negative widths- indicating they might be in the wrong cluster. Nonetheless, the negative scores are very small and should not be too much cause for concern.

Now the silhouette method for PAM:

```
pam.sil<-silhouette(pam_legprof$cluster, dist=legprof_dist)
fviz_silhouette(pam.sil)
```

```
##   cluster size ave.sil.width
## 1       1   42          0.70
## 2       2    7          0.14
```

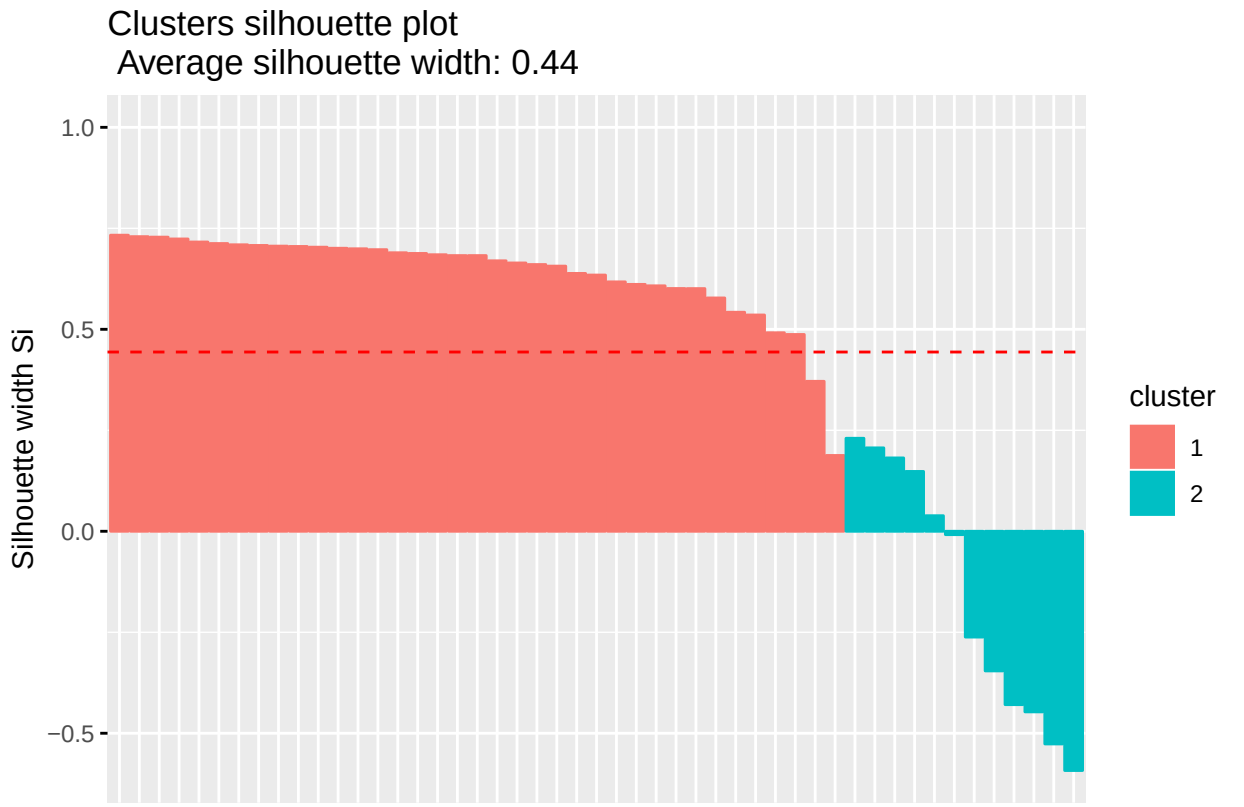


Similar to the K-Means results, the first and second cluster have high and low scores respectively. However, there is one large negative width, indicating possible misallocation.

Finally, for GMM:

```
library(factoextra, quietly=TRUE)
gmm.sil<-silhouette(gmm_leg_cluster, dist=legprof_dist)
fviz_silhouette(gmm.sil)
```

```
##   cluster size ave.sil.width
## 1       1   37          0.64
## 2       2   12         -0.15
```

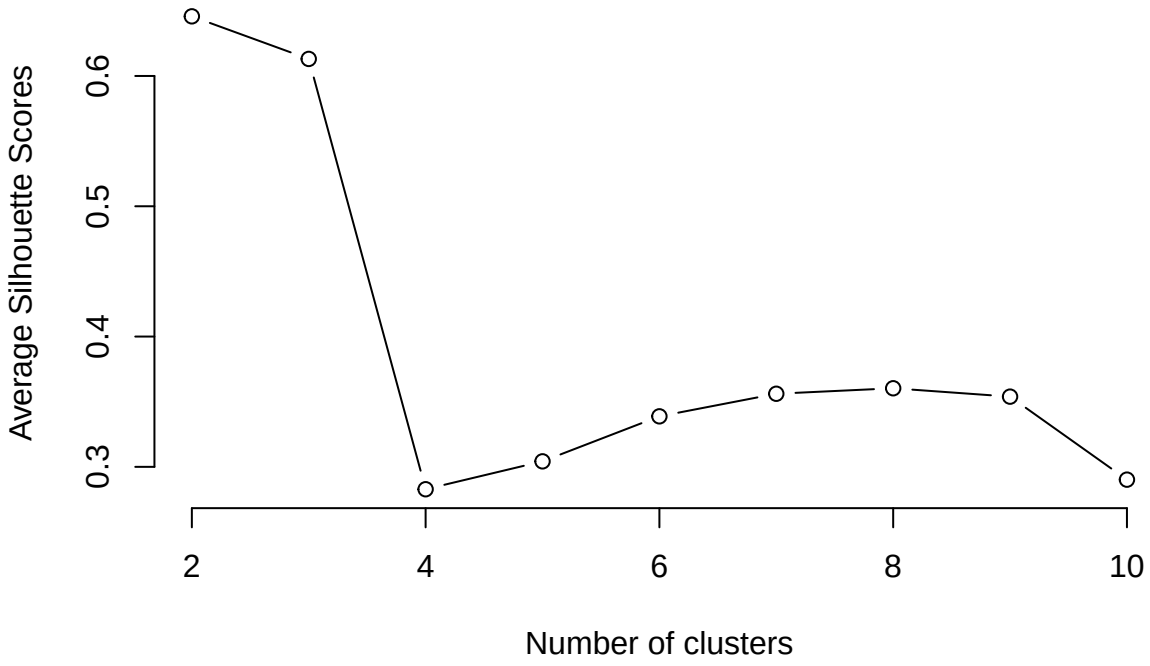


GMM performs the worst of the three methods, with several negative silhouette widths. These indicate that many states may have been placed in the wrong silhouette entirely. Furthermore, the overall average silhouette width is the lowest for the 3 methods.

Based on these three methods, it is clear that the closest fit (with the highest average silhouette width is the K-Means approach.

Now let us check what would be the best value of k for kmeans using silhouette scores. We calculate this average width for several values of k.

```
silhouette_score <- function(k){
  km <- kmeans(legprof_sub, centers = k, nstart=25)
  ss <- silhouette(km$cluster, legprof_dist)
  mean(ss[, 3])
}
k <- 2:10
avg_sil <- sapply(k, silhouette_score)
plot(k, type='b', avg_sil, xlab='Number of clusters', ylab='Average Silhouette Scores', frame=FALSE)
```



Clearly, the best score lies at 2 clusters.

10. Discuss the validation output.

a. What can you take away from the fit? As we realized from the clusterabiliy analysis, there are essentially two groups of states, one closely located, and the other more scattered.

b. Which approach is optimal? And optimal at what value of k? K-Means appears to be the best approach, given the silhouette score. Also, unlike the other two methods, it does not have any substantial negative silhouette scores, suggesting largely 'correct' allocation. As we saw in the plot above, the best fit is at k=2.

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

As researchers, we may already be aware of certain groupings within our data set based on prior domain expertise. For example, we may know that Washington DC is an outlier in terms of high voter preference for the Democratic Party in the US. This one outlier alone can skew results, and suggest the creation of a third cluster even when in terms of political science, this is uncalled for (since Washington DC is not a federal state).

Furthermore, the choice of variables used for clustering pays a key role. As we saw earlier, the highly correlated variables of session length and total session length could be reduced to one for improved parsimony of the model.Correlated variables may be affecting our choice of method. In this case, a method may seem optimal only due to a lack of exploratory analysis, as emphasized by statisticians like John Tukey.

Furthermore, if our relationship of interest in this data was between total session length and expenditure (as plotted for Q8), we may prefer a method like GMM that includes all the seeming outliers from the thick cluster in the lower left corner. These variables may not contribute the largest information to the dataset, and thus not change the 'optimal' partitioning method. Nonetheless, they would be favoured due to the question that the researcher is trying to address.

Clustering is only as 'good' as the data provided to it, and its methods are only as 'relevant' as the researchers' interests permit. That being said, the range of rigorous statistical methods metrics ensure that concessions only be made to an acceptable deviation from optimal methods.