# Problem Set 3 - Allison Collins

*Allison Collins*

*10/23/2019*

### 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.

```
load("~/Documents/PS3/State Leg Prof Data & Codebook/legprof-components.v1.0.RData")

#yields a df named "x", let's rename
prof <-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)

```
#Select desired columns + 2009/10 session
prof <- subset(prof, sessid == '2009/10', select = c("stateabv", "t_slength", "slength", "salary_real",

#Remove missing data
prof = prof[complete.cases(prof), ]

#Store names
prof_names <- subset(prof, select = "stateabv")

#Prior to scaling the data, let's store the summary stats so that we will be able to see actual means, 
summary <- summary(prof)

#Drop names + scale the numeric columns
prof <- scale(subset(prof, select = c("t_slength", "slength", "salary_real", "expend")))
```

### Question 3:

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

Now, let's look at summary statistics for the newly scaled data

```
summary
```

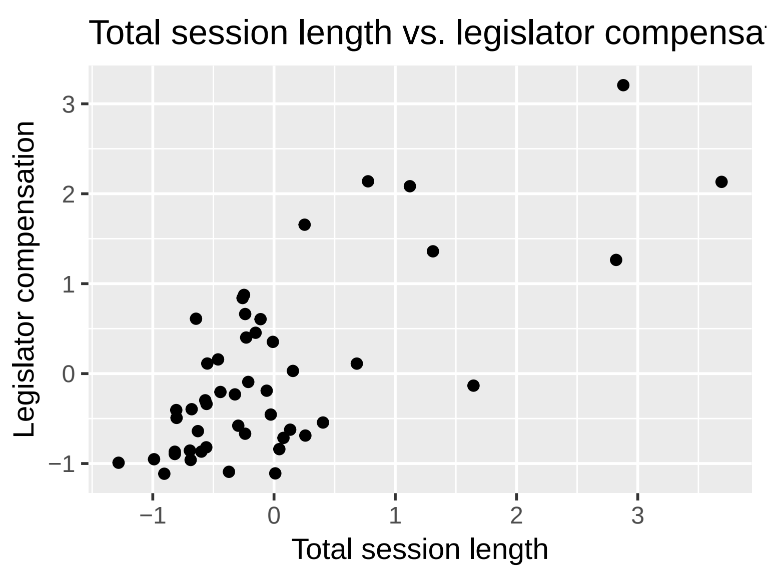
```
##    stateabv            t_slength          slength        salary_real
```

```
##   Length:49          Min.   : 40.00   Min.   : 40.0   Min.   :  0.00
##   Class :AsIs        1st Qu.: 97.42   1st Qu.: 93.0   1st Qu.: 19.69
##   Mode  :character   Median :127.77   Median :123.0   Median : 40.33
##                      Mean   :147.80   Mean   :138.5   Mean   : 54.99
##                      3rd Qu.:159.00   3rd Qu.:151.2   3rd Qu.: 77.43
##                      Max.   :458.15   Max.   :427.1   Max.   :213.41
##       expend
##   Min.   :  70.43
##   1st Qu.: 277.08
##   Median : 535.14
##   Mean   : 744.47
##   3rd Qu.: 724.91
##   Max.   :5523.10
```

Per above, we can see that the average (mean) total session length is 147 days, which is approximately just under 10 days higher than the regular session length. In both regular and total session length, the median is lower than the mean, suggesting some long sessions may be skewing the data.

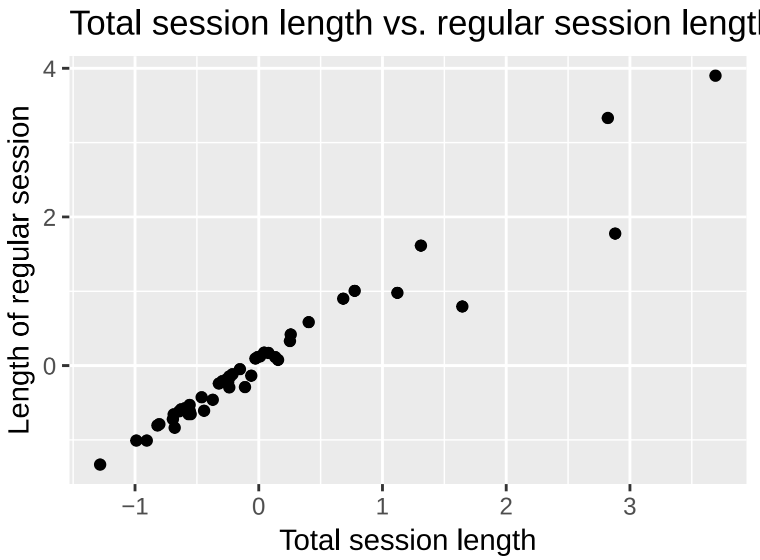
Salary figures show that while there are some state(s) as discussed in class that are "volunteer" and thus make 0, it ranges all the way up to 213 thousand dollars (and is right skewed, as the mean is higher than the median again). Expenditures also range, and are again skewed rightward as the mean is in the 700s with median about 200 dollars lower.

```
par(mfrow = c(3,2))

df<-as.data.frame(prof)

df %>%
  ggplot(aes(x = t_slength, y = salary_real)) +
  geom_point() +
  labs(x = "Total session length",
       y = "Legislator compensation",
       title = "Total session length vs. legislator compensation")
```

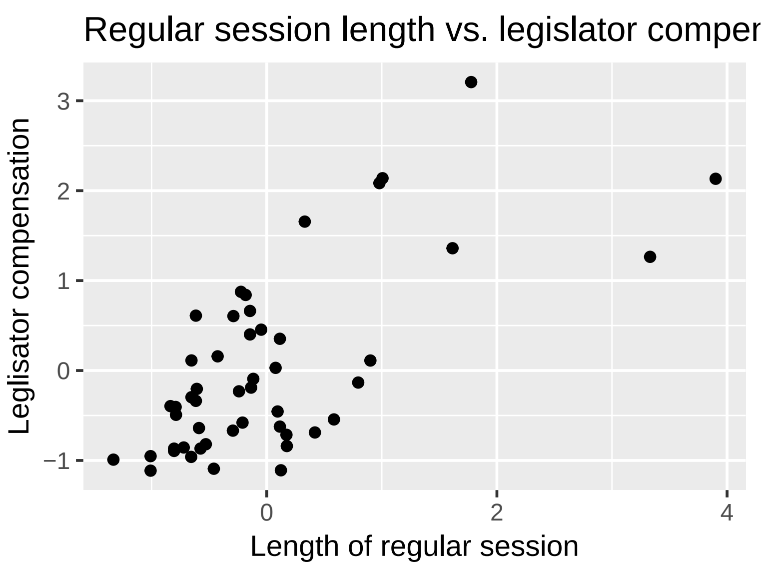



```
df %>%
  ggplot(aes(x = t_slength, y = slength)) +
  geom_point() +
  labs(x = "Total session length",
```

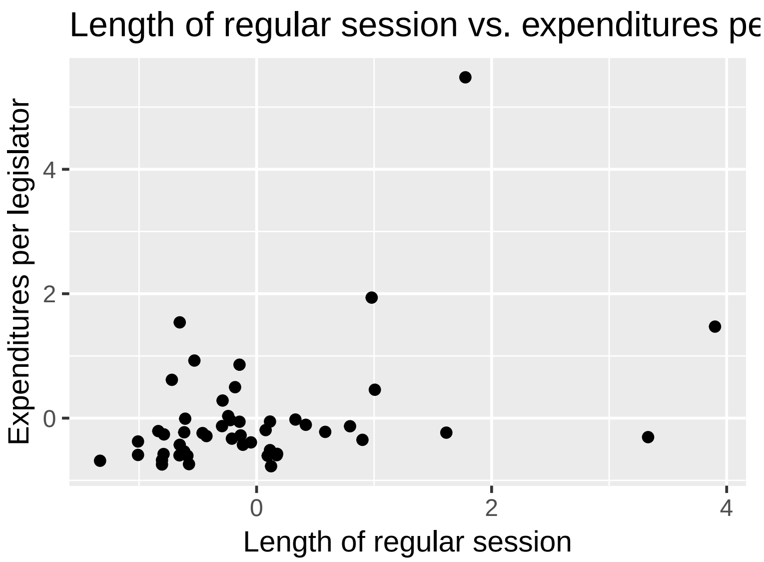
```
       y = "Length of regular session",
       title = "Total session length vs. regular session length")
```



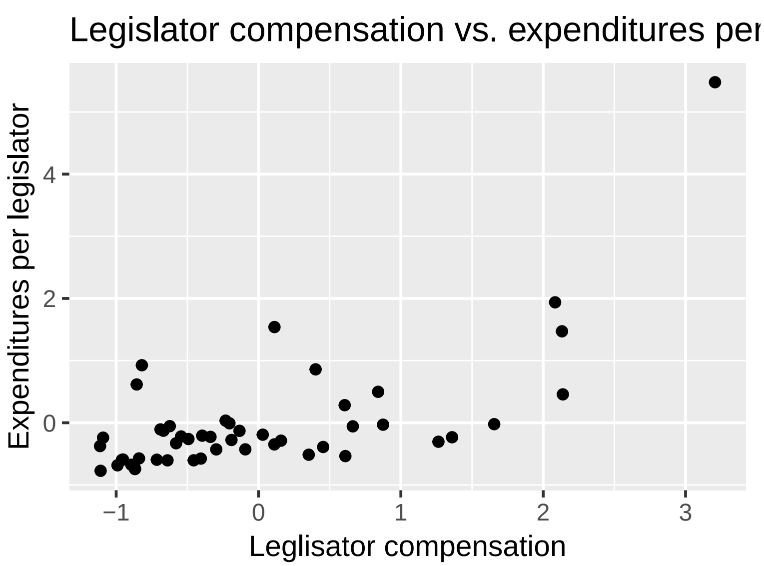

```
df %>%
  ggplot(aes(x = slength, y = salary_real)) +
  geom_point() +
  labs(x = "Length of regular session",
       y = "Leglisator compensation",
       title = "Regular session length vs. legislator compensation")
```



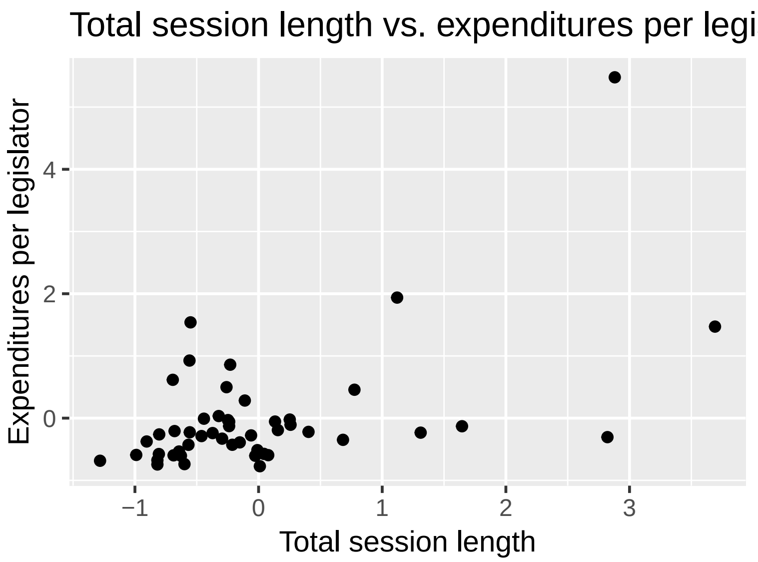

```
df %>%
  ggplot(aes(x = slength, y = expend)) +
  geom_point() +
  labs(x = "Length of regular session",
       y = "Expenditures per legislator",
       title = "Length of regular session vs. expenditures per legislator")
```




```
df %>%
  ggplot(aes(x = salary_real, y = expend)) +
  geom_point() +
  labs(x = "Leglisator compensation",
       y = "Expenditures per legislator",
       title = "Legislator compensation vs. expenditures per legislator")
```




```
df %>%
  ggplot(aes(x = t_slength, y = expend)) +
  geom_point() +
  labs(x = "Total session length",
       y = "Expenditures per legislator",
       title = "Total session length vs. expenditures per legislator")
```

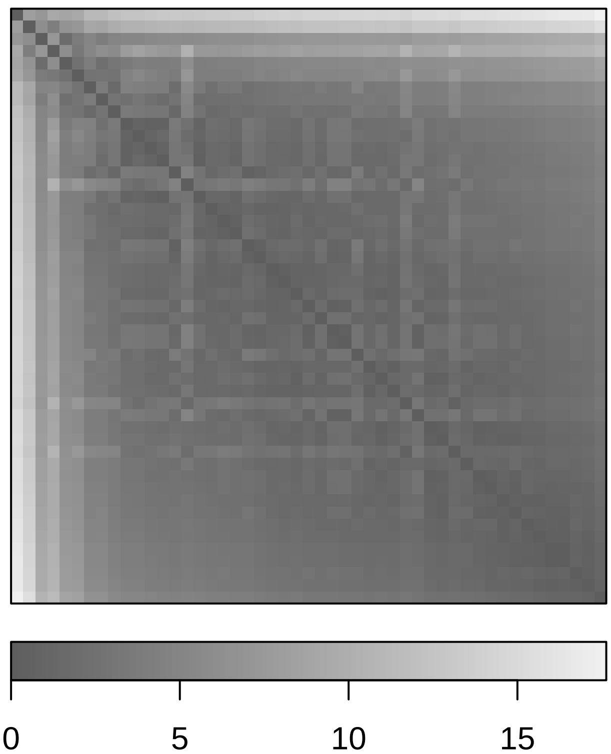



From the above scatter plots we can see that there appears to be a correlation between total and regular session length, which intuitively makes sense. In addition, these seem to correlate with the legislator compensation. There appears to be less of a relationship with expenditures.

**Question 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.

```r
#Calculate distance using our scaled df
prof_dist <- dist(prof, method = "manhattan")
dissplot(prof_dist)
```




In the above ODI, there may be few squares emerging (e.g. one along diagonal, bottom right), but overall there is less clear delineation than we saw in class – so I do not have as much confidence in the clusterability of the data (as I did with for example the old faithful data we used on the last homework assignment).

### 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.

```
kmeans <- kmeans(prof,
                 centers = 2,
                 nstart = 15)
str(kmeans)
```

```
## List of 9
##  $ cluster     : Named int [1:49] 1 1 1 1 2 1 1 1 1 1 ...
##   ..- attr(*, "names")= chr [1:49] "19" "38" "57" "76" ...
##  $ 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"
```

```r
#let's save cluster assignments (add to the state names)
prof_names$k_cluster = as.factor(kmeans$cluster)
```

From the above, we can see that 6 states were put into one cluster, and the remaining 43 were placed into the second cluster. Specifically, we can check (as below) and discover that the states that were in that cluster were PA, OH, NYC, MA, MI and CA.

```r
which(prof_names$cluster ==1)
```

```
## integer(0)
```

```r
prof_names[5,]
```

```
##    stateabv k_cluster
## 95       CA         2
```

```r
prof_names[21,]
```

```
##     stateabv k_cluster
## 399       MA         2
```

```r
prof_names[22,]
```

```
##     stateabv k_cluster
## 418       MI         2
```

```r
prof_names[32,]
```

```
##     stateabv k_cluster
## 608       NY         2
```

```r
prof_names[35,]
```

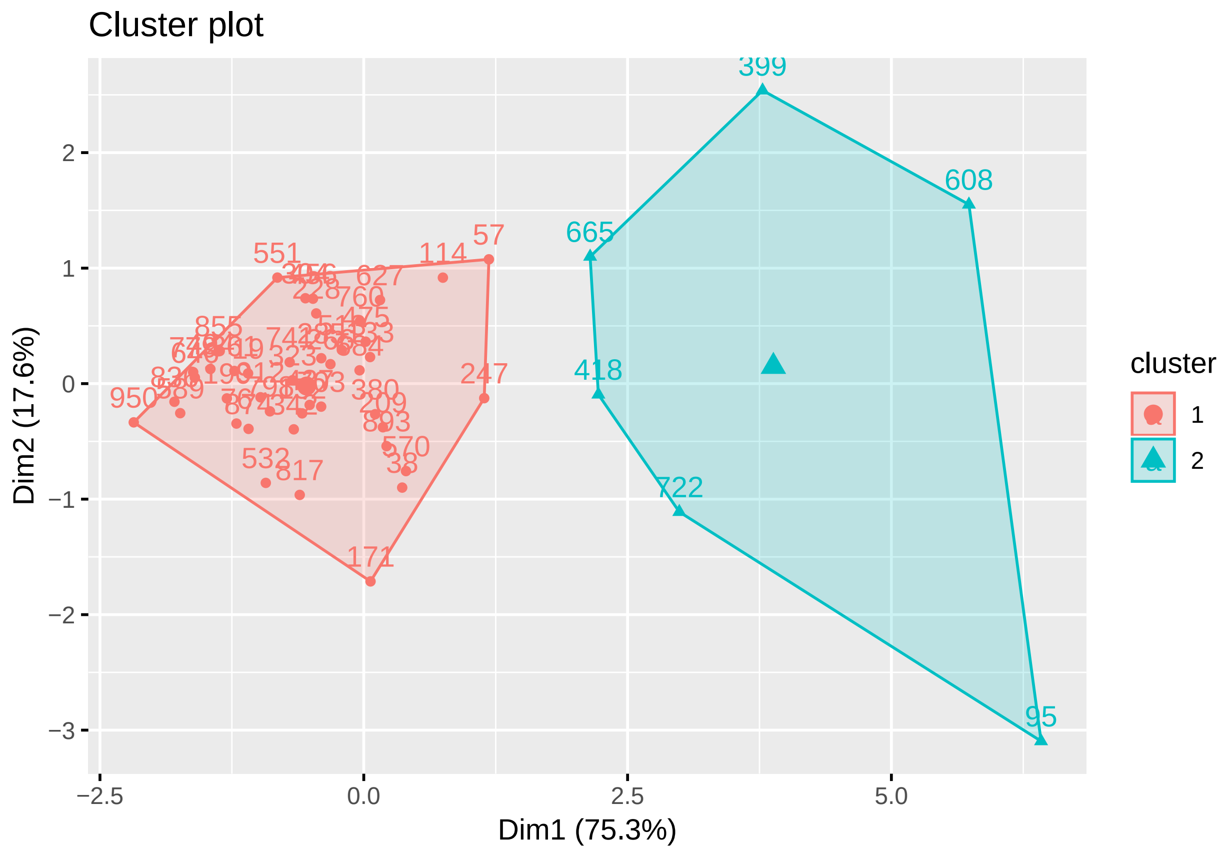
```
##     stateabv k_cluster
## 665       OH         2
```

```r
prof_names[38,]
```

```
##     stateabv k_cluster
## 722       PA         2
```

We can also visualize the outcome of this clustering (since we have more than two dimensions, components will be used)

```r
fviz_cluster(kmeans, prof)
```




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

```
set.seed(1234)
gmm1 <- mvnormalmixEM(prof, k = 2)
```

```
## number of iterations= 20
```

```
gmm1$mu
```

```
## [[1]]
## [1] -0.3225760 -0.3008868 -0.3158336 -0.3566484
##
## [[2]]
## [1] 0.9567777 0.8924463 0.9367794 1.0578383
```

```
gmm1$sigma
```

```
## [[1]]
##           [,1]       [,2]       [,3]       [,4]
## [1,] 0.1870505 0.20906090 0.10530159 0.04208450
## [2,] 0.2090609 0.24042904 0.11260289 0.03538731
## [3,] 0.1053016 0.11260289 0.40683486 0.08685191
```

```
## [4,] 0.0420845 0.03538731 0.08685191 0.06580480
##
## [[2]]
##           [,1]      [,2]     [,3]      [,4]
## [1,] 2.1062552 2.0100844 1.271799 0.6272022
## [2,] 2.0100844 2.1070027 1.195474 0.2039618
## [3,] 1.2717987 1.1954741 1.504998 1.0097881
## [4,] 0.6272022 0.2039618 1.009788 2.1936351
```

```
gmm1$lambda
```

```
## [1] 0.7478602 0.2521398
```

```
posterior <- data.frame(cbind(gmm1$x, gmm1$posterior))
posterior$component <- ifelse(posterior$comp.1 > 0.5, 1, 2)
#posterior - used to examine probabilities; eliminating in knit pdf due to length
table(posterior$component)
```

```
##
##  1  2
## 37 12
```

```
#add to the df where we are collating the cluster assignments for later
prof_names$gmm_cluster <- as.factor(posterior$component)
```

As we can see, we end up with one cluster that has 37 states and one which has 12 here, which is different than what we did kmeans. In addition, the probability scores of belonging to each component (cluster) here are very high or low (have commented out now due to length as above). While Gaussian models are a form of soft partitioning and thus it might be possible to belong to more than one cluster, here there seem to be very high or low probabilities for belonging to a cluster.

### Question 7

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

```
fcm1 <- fcm(prof, centers=2)

#let's add in the cluster assignments to the names dataframe
prof_names$fuzzy_cluster = as.factor(fcm1$cluster)

#Let's print out selective output
fcm1$cluster
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
##  1  1  1  1  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  2  1  1  1
## 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
##  1  1  1  1  1  1  2  1  1  2  1  1  2  1  1  1  1  1  1  1  1  1  1  1
```

```
fcm1$csize
```

```
##  1  2
## 43  6
```

```r
fcm1$u
```

```
##     Cluster 1   Cluster 2
## 19  0.9721473 0.027852666
## 38  0.8640894 0.135910554
## 57  0.5710701 0.428929853
## 76  0.9791492 0.020850785
## 95  0.2608918 0.739108167
## 114 0.7433790 0.256620995
## 133 0.9572892 0.042710755
## 152 0.9381251 0.061874931
## 171 0.8201328 0.179867154
## 190 0.9751804 0.024819583
## 209 0.8874955 0.112504477
## 228 0.9730065 0.026993537
## 247 0.5929603 0.407039736
## 266 0.9911745 0.008825492
## 285 0.9899603 0.010039750
## 304 0.9555633 0.044436705
## 323 0.9922112 0.007788783
## 342 0.9913693 0.008630717
## 361 0.9817922 0.018207834
## 380 0.9196783 0.080321698
## 399 0.1737842 0.826215850
## 418 0.2299977 0.770002321
## 437 0.9839480 0.016051988
## 456 0.9580221 0.041977931
## 475 0.9353843 0.064615676
## 494 0.9685935 0.031406529
## 513 0.9639541 0.036045899
## 532 0.9433932 0.056606823
## 551 0.9397234 0.060276612
## 570 0.8488997 0.151100257
## 589 0.9499852 0.050014840
## 608 0.1444114 0.855588628
## 627 0.8923308 0.107669171
## 646 0.9624140 0.037586019
## 665 0.2302067 0.769793335
## 684 0.9445386 0.055461408
## 703 0.9948507 0.005149344
## 722 0.1293823 0.870617717
## 741 0.9947850 0.005215044
## 760 0.9257607 0.074239275
## 779 0.9611745 0.038825495
## 798 0.9928692 0.007130755
## 817 0.9203580 0.079641972
## 836 0.9508466 0.049153368
## 855 0.9709459 0.029054140
## 874 0.9818445 0.018155454
## 893 0.9069106 0.093089417
## 912 0.9895120 0.010487999
## 950 0.9256336 0.074366354
```

From above we can see that we end up with the same number within clusters as we do when we did kmeans and Gaussian (43 in one cluster, and 6 in the other). Additionally, skimming the probabilities (which again seem to overall be extremely high e.g. in the 90s and low for the non-assigned cluster - did not include due to length of output) might suggest that there were fewer cases of points being edge cases for inclusion in multiple clusters.

### Question 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)

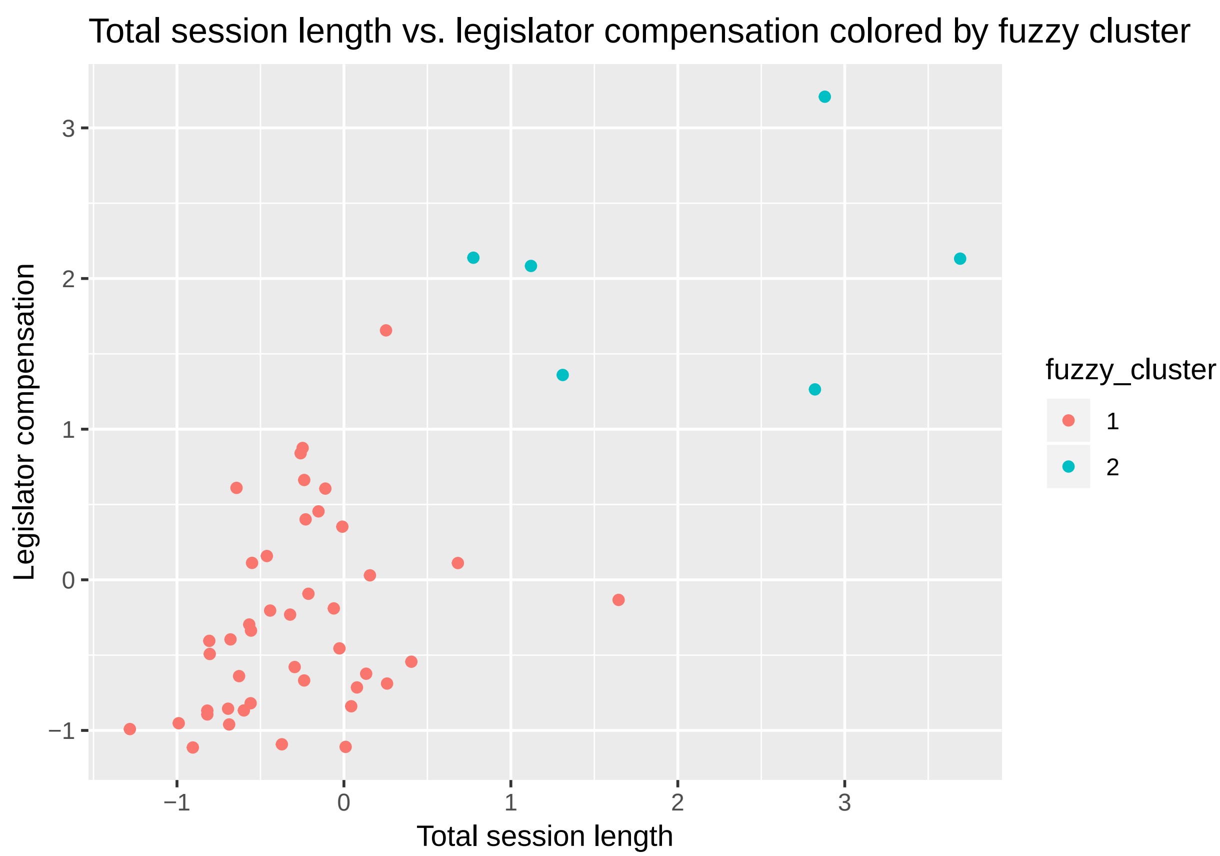
```r
new_df <- cbind(prof_names, prof)
new_df
```

```
##     stateabv k_cluster gmm_cluster fuzzy_cluster    t_slength     slength
## 19        AL         1           1             1 -0.371659901 -0.45947229
## 38        AK         1           2             1 -0.229408926 -0.14523091
## 57        AZ         1           2             1  1.645306675  0.79519547
## 76        AR         1           1             1 -0.803646214 -0.78817557
## 95        CA         2           2             2  2.880725686  1.77670986
## 114       CO         1           1             1  0.682733830  0.90088869
## 133       CT         1           1             1  0.155953294  0.07723846
## 152       DE         1           1             1 -0.643316513 -0.61557933
## 171       FL         1           2             1 -0.550306302 -0.65382896
## 190       GA         1           1             1 -0.806381805 -0.79128419
## 209       HI         1           1             1 -0.247368734 -0.22362235
## 228       ID         1           1             1 -0.026736988  0.09467375
## 247       IL         1           1             1  0.252174862  0.33052374
## 266       IN         1           1             1 -0.060515652 -0.13563473
## 285       IA         1           1             1 -0.212519663 -0.11644243
## 304       KS         1           1             1  0.043793814  0.17482220
## 323       KY         1           1             1 -0.238210429 -0.29390430
## 342       LA         1           1             1 -0.441952865 -0.60746990
## 361       ME         1           1             1 -0.627973378 -0.58854782
## 380       MD         1           1             1 -0.237853581 -0.14523091
## 399       MA         2           2             2  2.821493958  3.33129222
## 418       MI         2           2             2  0.775506248  1.00631164
## 437       MN         1           1             1 -0.461458846 -0.42635872
## 456       MS         1           1             1  0.078048154  0.17144321
## 475       MO         1           1             1 -0.009490831  0.11427161
## 494       MT         1           1             1 -0.687442853 -0.65612661
## 513       NE         1           1             1  0.133235911  0.11427161
## 532       NV         1           2             1 -0.693865590 -0.72100229
## 551       NH         1           1             1  0.010371952  0.12332714
## 570       NJ         1           2             1 -0.259262629 -0.18307507
## 589       NM         1           1             1 -0.904982206 -1.00888793
## 608       NY         2           2             2  3.691294567  3.90071117
## 627       NC         1           1             1  0.403940989  0.58407939
## 646       ND         1           1             1 -0.818275700 -0.80479995
## 665       OH         2           2             2  1.310731529  1.61452076
## 684       OK         1           1             1 -0.152217572 -0.04791749
## 703       OR         1           1             1 -0.322300309 -0.24119288
## 722       PA         2           2             2  1.120429207  0.97928013
```

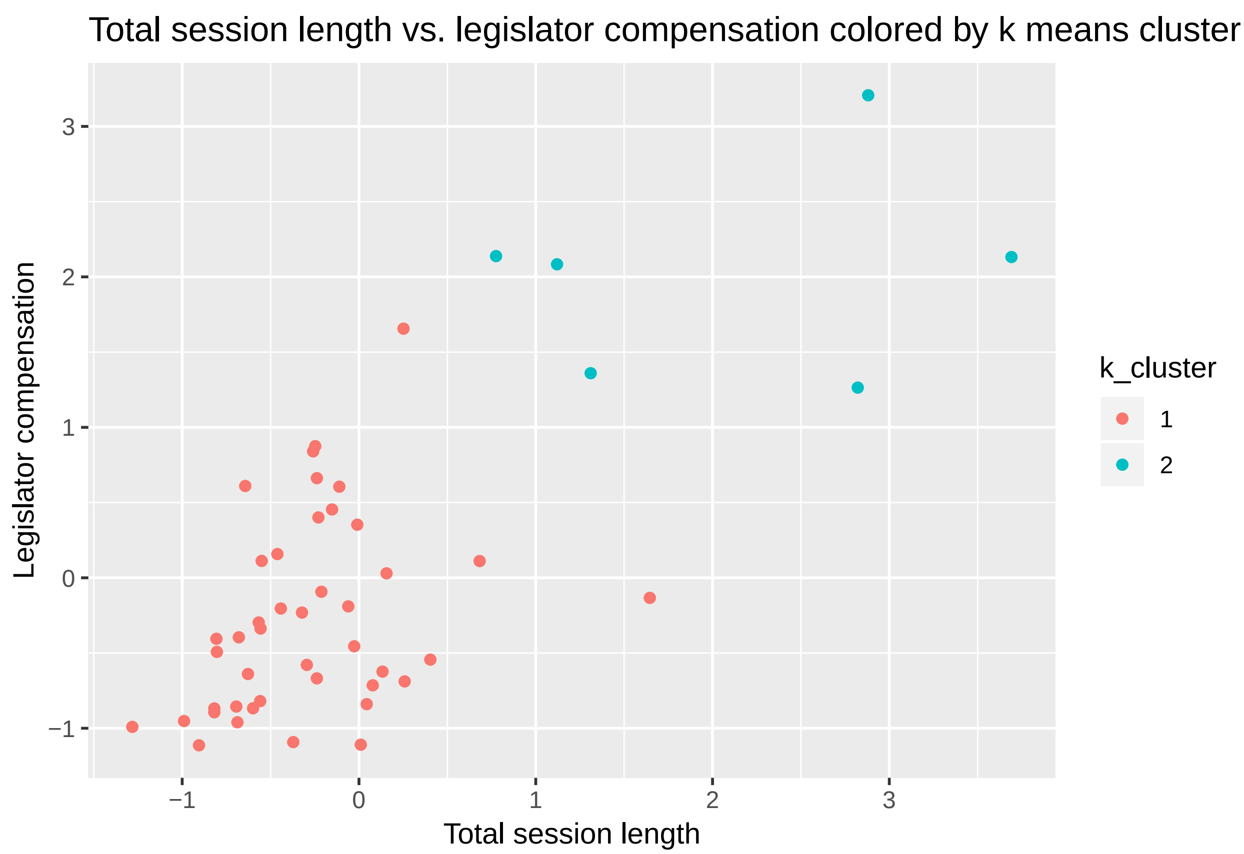
```
## 741       RI         1           1             1 -0.294944314 -0.21010659
## 760       SC         1           1             1  0.258478612  0.41878162
## 779       SD         1           1             1 -0.818275700 -0.80479995
## 798       TN         1           1             1 -0.556610007 -0.61557933
## 817       TX         1           2             1 -0.558750912 -0.52907846
## 836       UT         1           1             1 -0.989428844 -1.00888788
## 855       VT         1           1             1 -0.599190173 -0.57503206
## 874       VA         1           1             1 -0.679355001 -0.83615650
## 893       WA         1           1             1 -0.111183625 -0.28917375
## 912       WV         1           1             1 -0.567195612 -0.65382896
## 950       WY         1           1             1 -1.282137610 -1.33191452
##     salary_real       expend
## 19  -1.09200089 -0.239991004
## 38   0.40113330  0.859119849
## 57  -0.13356561 -0.129940807
## 76  -0.49239021 -0.261206056
## 95   3.20699144  5.478545259
## 114  0.11135948 -0.348553011
## 133  0.02971780 -0.191902436
## 152  0.61016386 -0.535852545
## 171  0.11213913  1.539500636
## 190 -0.40535082 -0.576530958
## 209  0.87503597 -0.030968419
## 228 -0.45539720 -0.604391173
## 247  1.65585716 -0.022426587
## 266 -0.19004288 -0.277057831
## 285 -0.09274477 -0.428020271
## 304 -0.83953996 -0.574076126
## 323 -0.66821257 -0.127784042
## 342 -0.20424551 -0.008731385
## 361 -0.63936779 -0.604554127
## 380  0.66244097 -0.057590394
## 399  1.26402392 -0.305301736
## 418  2.13811467  0.456899531
## 437  0.15793203 -0.287914432
## 456 -0.71457282 -0.594888550
## 475  0.35281484 -0.514269618
## 494 -0.96022656 -0.597593000
## 513 -0.62341582 -0.054489563
## 532 -0.85573358  0.616190643
## 551 -1.10918394 -0.772769645
## 570  0.84033067  0.498016486
## 589 -1.11326602 -0.374895524
## 608  2.13199154  1.471428802
## 627 -0.54377433 -0.220619983
## 646 -0.89381810 -0.675224373
## 665  1.35982443 -0.233023694
## 684  0.45425467 -0.389504090
## 703 -0.23104580  0.033991598
## 722  2.08360490  1.937703837
## 741 -0.57896192 -0.329505886
## 760 -0.68872917 -0.106730457
## 779 -0.86834092 -0.744838524
## 798 -0.33730246 -0.227228690
```

```
## 817 -0.81935590  0.925725780
## 836 -0.95212512 -0.591232328
## 855 -0.86728615 -0.739004684
## 874 -0.39561825 -0.207959325
## 893  0.60553675  0.282936858
## 912 -0.29684902 -0.428715471
## 950 -0.99080347 -0.684772532
```

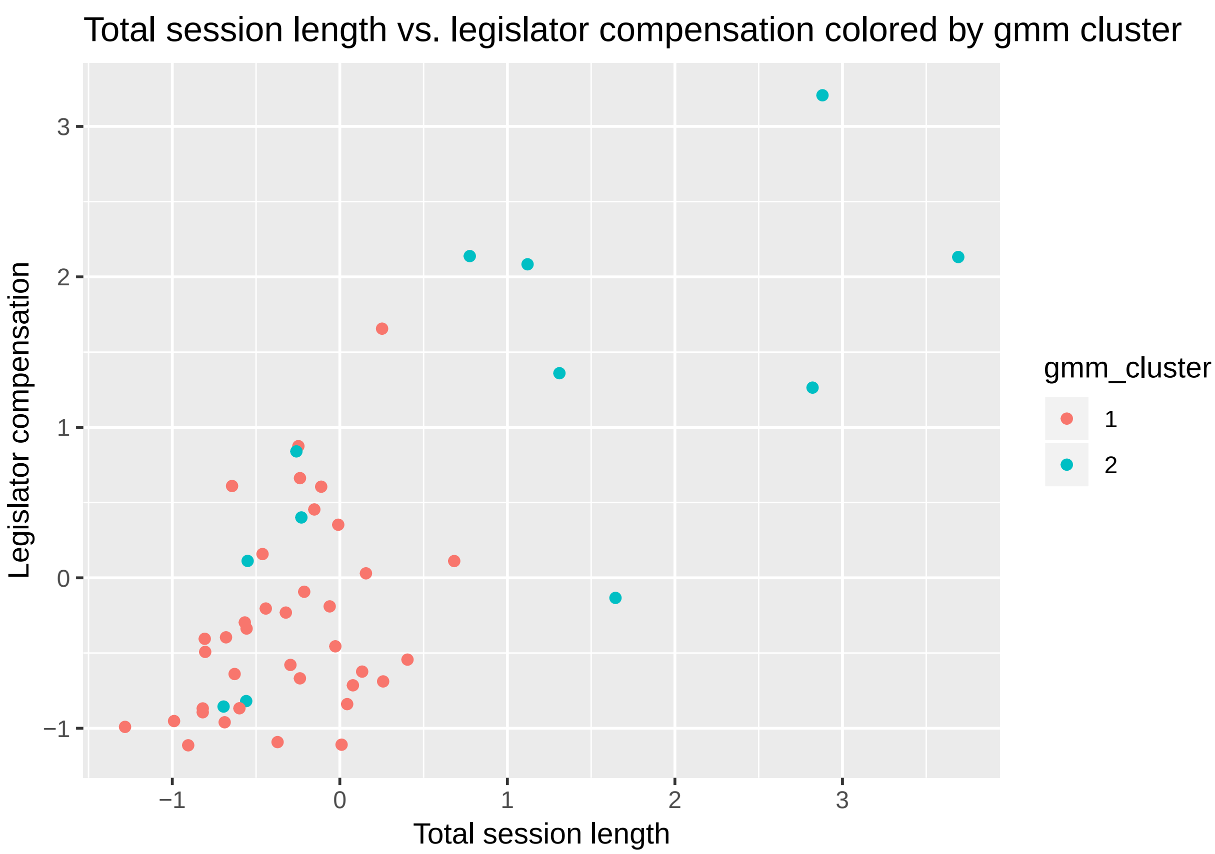
```r
new_df %>%
  ggplot(aes(x = t_slength, y = salary_real, color = fuzzy_cluster)) +
  geom_point() +
  labs(x = "Total session length",
       y = "Legislator compensation",
       title = "Total session length vs. legislator compensation colored by fuzzy cluster")
```




```r
new_df %>%
  ggplot(aes(x = t_slength, y = salary_real, color = k_cluster)) +
  geom_point() +
  labs(x = "Total session length",
       y = "Legislator compensation",
       title = "Total session length vs. legislator compensation colored by k means cluster")
```




```r
new_df %>%
  ggplot(aes(x = t_slength, y = salary_real, color = gmm_cluster)) +
  geom_point() +
  labs(x = "Total session length",
       y = "Legislator compensation",
       title = "Total session length vs. legislator compensation colored by gmm cluster")
```




I chose to look at legislator compensation and total session length to compare how the "professionalism" clusters stacked up across the different clustering mechanisms. As we can see above, the kmeans and fuzzy c means classified the professionalism clusters the same way (albeit, the color looks different since that is random each time). Both suggest that there are groupings emerging with longer session length and higher compensation vs. lower compensation and shorter total session length.

The Gaussian model has a few additional points within the lower compensation / session length that appear to be grouped with the higher compensation / higher total session length. Of course, if we had used two different variables to plot, we might see slightly different results - thus looking at just these two features is not creating a complete picture.

In sum, if we feel that these are valid measures of "professionalism", it would seem that the clusters suggested by kmeans and fuzzy c-means do a better job of finding "alike" state legislatures on the particular dimensions examined.

### Question 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)

```
#Validate for kmeans, GMM ("model" according to documentation) and fuzzy C means ("fanny" according to
internal <- clValid(prof, 2:5,
                    clMethods = c("kmeans","model","fanny"),
                    validation = "internal")
```

```
## Warning in vClusters(mat, clMethods[i], nClust, validation = validation, :
## fanny unable to find 5 clusters, returning NA for these validation measures
```

```r
summary(internal)
```

```
##
## Clustering Methods:
##  kmeans model fanny
##
## Cluster sizes:
##  2 3 4 5
##
## Validation Measures:
##                            2       3       4       5
##
## kmeans Connectivity   8.4460 10.8960 16.1885 28.7437
##        Dunn           0.1735  0.2581  0.2562  0.1090
##        Silhouette     0.6458  0.6131  0.4932  0.3042
## model  Connectivity  10.7393 28.6119 39.0687 67.8401
##        Dunn           0.1522  0.0633  0.0225  0.0258
##        Silhouette     0.6314  0.2588  0.1861  0.0085
## fanny  Connectivity  17.6123 28.4960 33.8143      NA
##        Dunn           0.0457  0.0479  0.0572      NA
##        Silhouette     0.3382  0.1648  0.2600      NA
##
## Optimal Scores:
##
##              Score  Method Clusters
## Connectivity 8.4460 kmeans 2
## Dunn         0.2581 kmeans 3
## Silhouette   0.6458 kmeans 2
```

Above, I display the performance on validation measures – since the validation package outputs several dimensions, I will include all three (Dunn, Silhouette, Connectivity). Kmeans is the strongest performer on all 3 (we want high values for silhouette width and the Dunn index and low values on connectivity)

### 10. Discuss the validation output.

## a. What can you take away from the fit?

We use internal validation to understand how well clustering algorithms perform relative to other algorithms or specifications – given that we don't have a label we can use to validate via an external measure (where we could see how well the model captures the "real world" like in supervised learning where we are predicting a label that we can then check classification against).

Thus we will compare across different numbers of clusters and types of clustering to see which is the best along the aforementioned dimensions in Q9 – and this may also require prioritizing which of the metrics we think are most important, as there is not necessarily a consistent best performer among Dunn, silhouette width, and connectivity.

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

In this specific case, kmeans was the strongest performer across all three measures of internal validation – however, while for connectivity and silhouette width, 2 clusters would be the optimal number, for Dunn index three clusters would be best.

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

Firstly, as above, we may not find a strongest performer on all dimensions and thus be forced to choose re: which measure we would like to prioritize. However, more broadly, it can also depend on what we want to do as a "takeaway" from the clustering. If, say, we were working at a firm looking at advertising campaigns and we only had budget to do two separate campaigns, we may elect to take the best clustering algorithm of k = 2 vs. one that suggests 5 clusters, because we cannot action that outcome. Thus, real world considerations on how we want to use the outcome could influence our choice.

Finally, we could also incorporate domain knowledge re: what number of clusters are commonly used in the field and/or could make a prioritization based on ease of explanation – both of which could contradict the "top performer" based on the above internal validation measures.