Problem_Set_3_Kim

Daejin Kim

10/25/2019

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.

```
#install.packages("tidyverse")
#install.packages("skimr")
#install.packages("ggplot2")
#install.packages("clustertend")
#install.packages("seriation")
library(magrittr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------------------- tidyvers
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------------------ tidyverse_confl
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x purrr::set_names() masks magrittr::set_names()
```

```
library(skimr)
```

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

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

```
library(dplyr)
library(ggplot2)
library(seriation)
```

```
## Registered S3 method overwritten by 'seriation':
##   method         from
##   reorder.hclust gclus
```

```r
library(clustertend)

library(miceadds)
```

```
## Loading required package: mice

## Loading required package: lattice

##
## Attaching package: 'lattice'

## The following object is masked from 'package:seriation':
##
##     panel.lines

##
## Attaching package: 'mice'

## The following object is masked from 'package:tidyr':
##
##     complete

## The following objects are masked from 'package:base':
##
##     cbind, rbind

## * miceadds 3.5-14 (2019-08-23 13:23:33)
```

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;

```r
sl_dta<-load.Rdata2("legprof-components.v1.0.RData", path=getwd())
head(sl_dta)
```

```
##   fips stateabv   state  sessid t_slength slength salary_real   expend
## 1    1       AL Alabama  1973/4     46.00    36.0    1.768022 125.0973
## 2    1       AL Alabama  1975/6    110.00    74.0    2.933038 203.8466
## 3    1       AL Alabama  1977/8     83.00    60.0    2.082810 184.0115
## 4    1       AL Alabama 1979/80     65.00    60.0    1.694951 175.9863
## 5    1       AL Alabama  1981/2    218.68   149.1    3.472914 204.1236
## 6    1       AL Alabama  1983/4    188.86   149.1   11.705946 206.6745
##   year       mds1        mds2
## 1 1974 -1.7061814  0.38482016
## 2 1976 -1.2128817 -0.08150655
## 3 1978 -1.4149657  0.12789978
## 4 1980 -1.5431539  0.27268842
## 5 1982 -0.5013642 -1.00387760
## 6 1984 -0.5840194 -0.74132360
```

```
sl_dta1<-filter(sl_dta, sessid=="2009/10")
sl_dta2<-select(sl_dta1, "t_slength","slength","salary_real","expend")
sl_dta3<- drop_na(sl_dta2)
sl_dta4<-scale(sl_dta3)
new_sldta<-data.frame(sl_dta4)
```

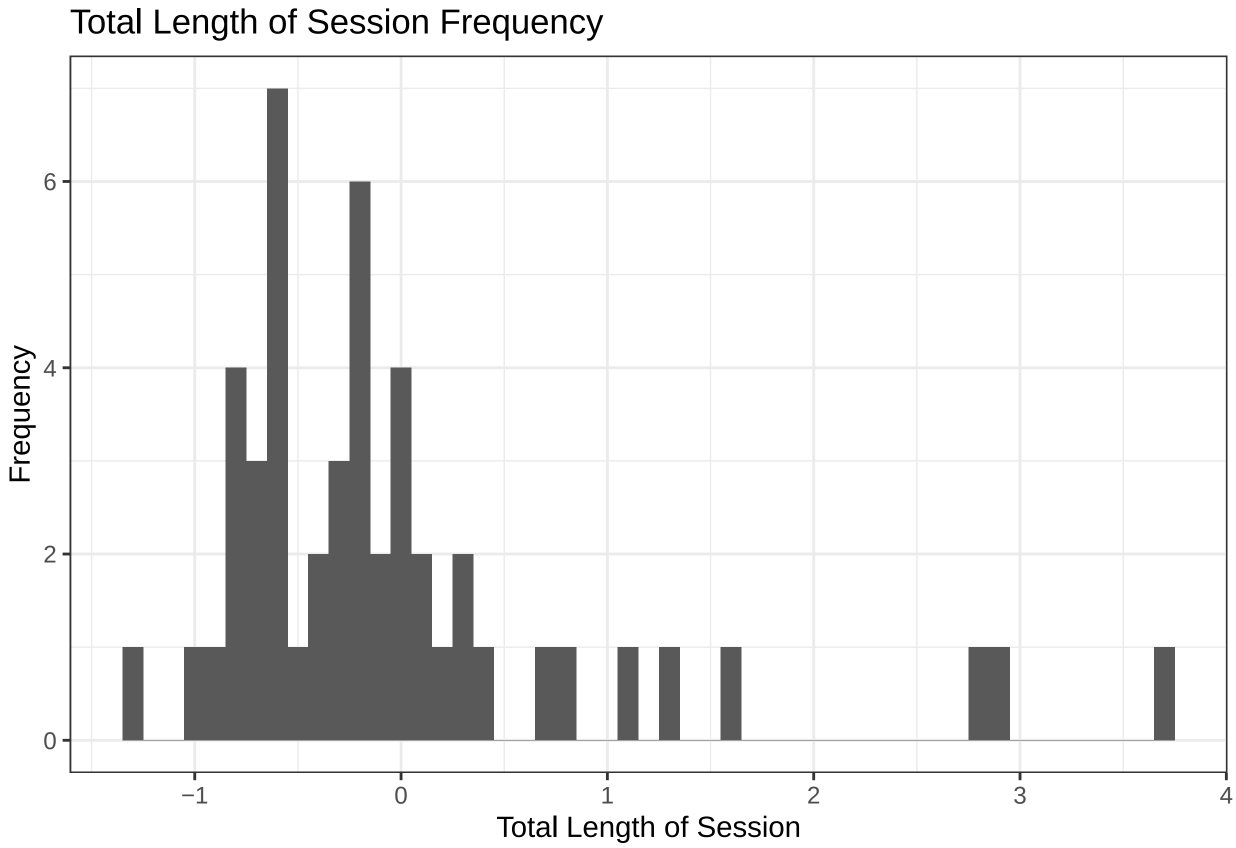
e. and anything else you think necessary to get this subset of data into workable form (hint: consider storing the state names as a separate object to be used in plotting later)

```
state<-filter(sl_dta, sessid=="2009/10") %>%
  select("state")
```

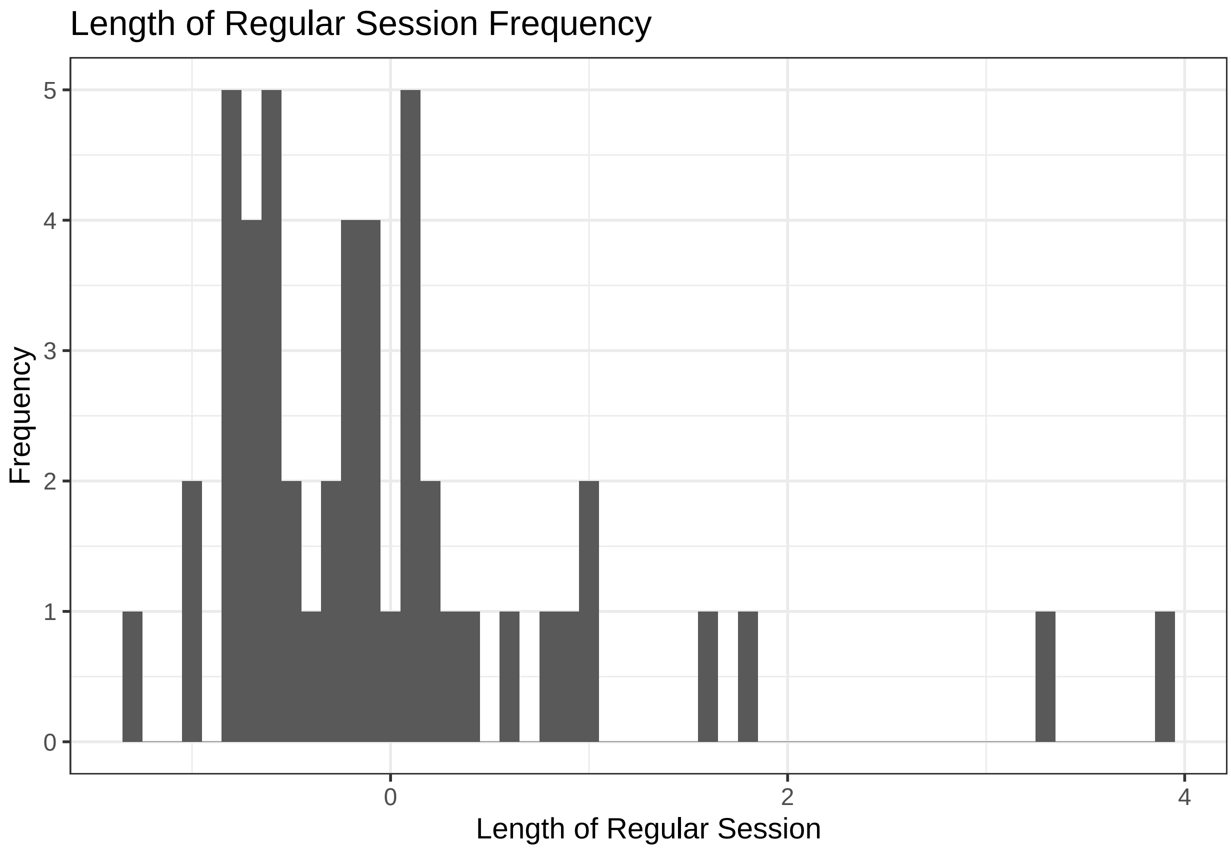
3. Perform quick EDA visually or numerically and discuss the patterns you see.

Across all the histograms for each variable, it seems that many states are centered around left end of the scale, which may suggest that they generally have short session, compensation, and expenditures compared to some outliers that are located on the right side of the graphs.

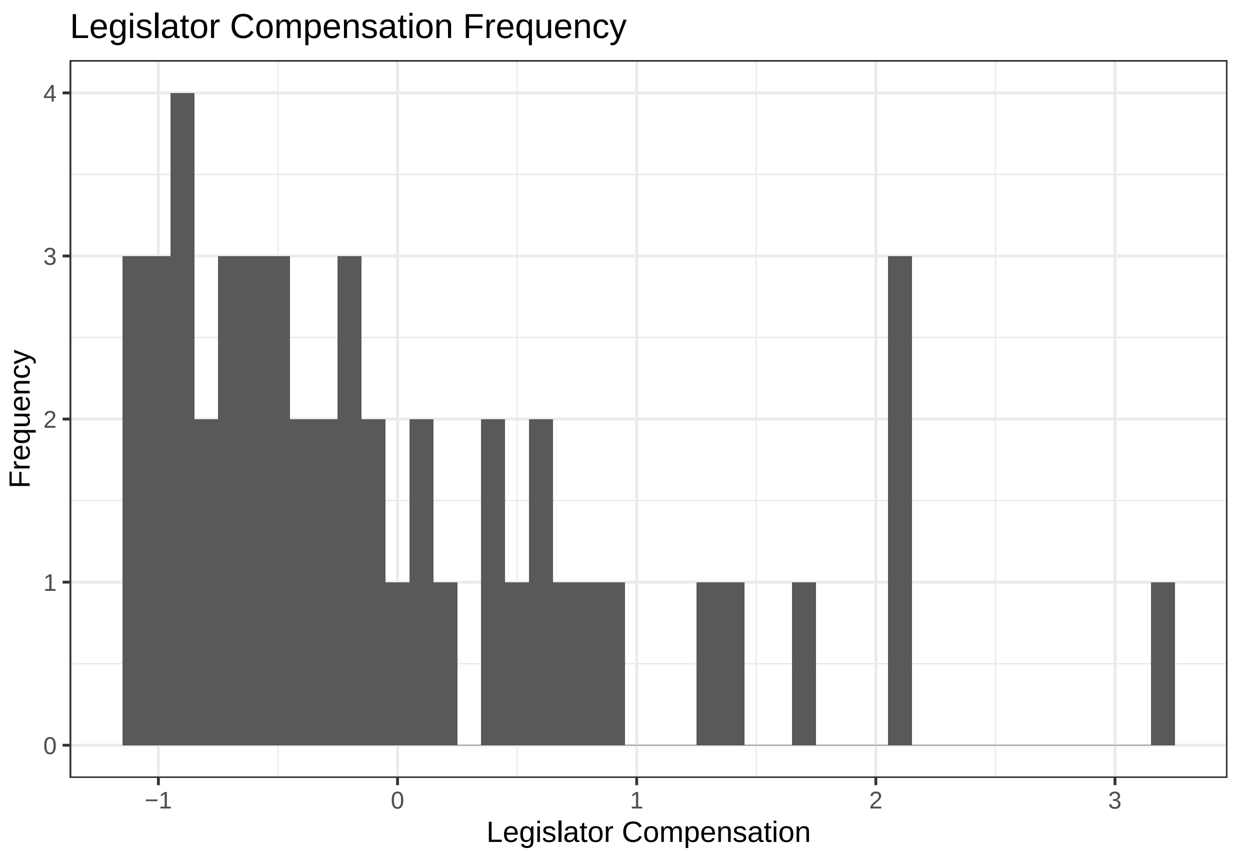
```
ggplot(data = new_sldta) +
  geom_histogram(aes(x = t_slength), binwidth = 0.1) +
  labs(x="Total Length of Session",
       y="Frequency",
       title="Total Length of Session Frequency") +
  theme_bw()
```



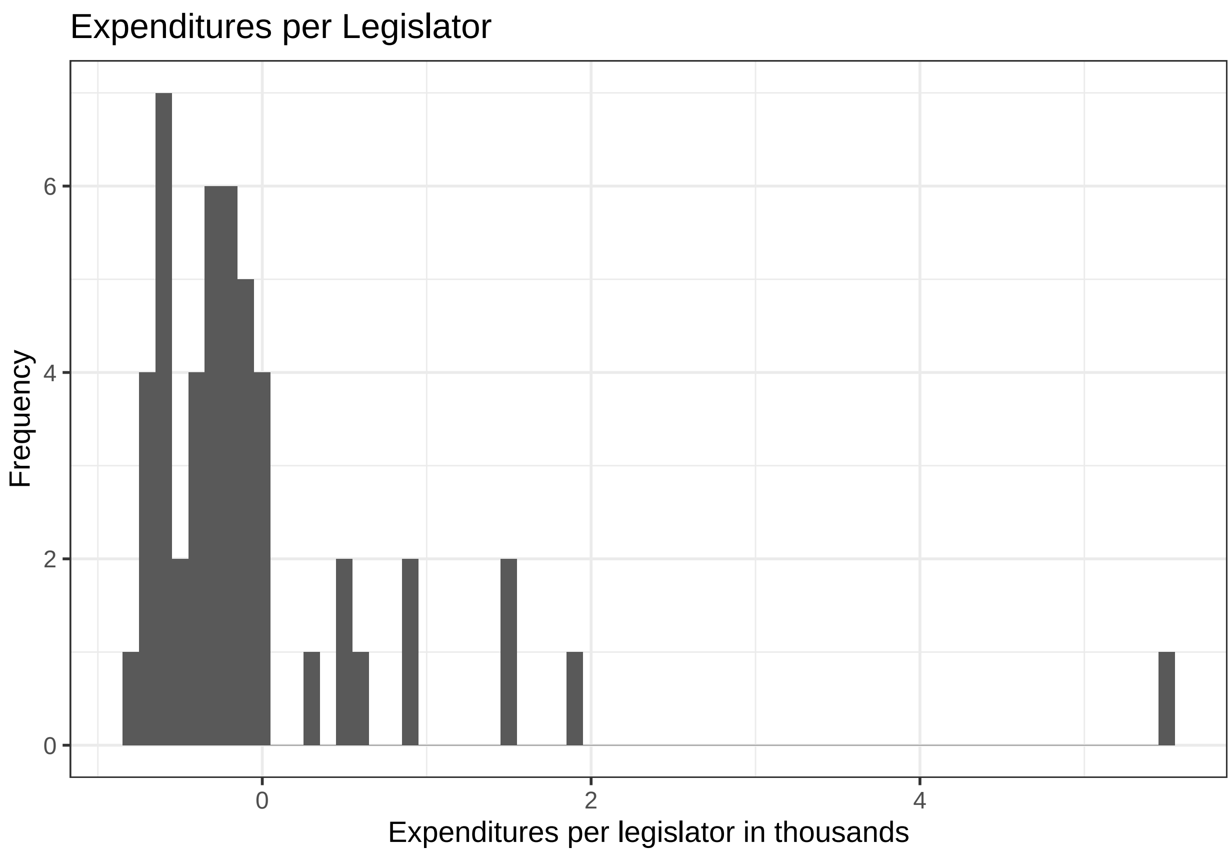
```
ggplot(data = new_sldta) +
  geom_histogram(aes(x = slength), binwidth = 0.1) +
  labs(x="Length of Regular Session",
       y="Frequency",
       title="Length of Regular Session Frequency") +
  theme_bw()
```



```
ggplot(data = new_sldta) +
  geom_histogram(aes(x = salary_real), binwidth = 0.1) +
  labs(x="Legislator Compensation",
       y="Frequency",
       title="Legislator Compensation Frequency") +
  theme_bw()
```



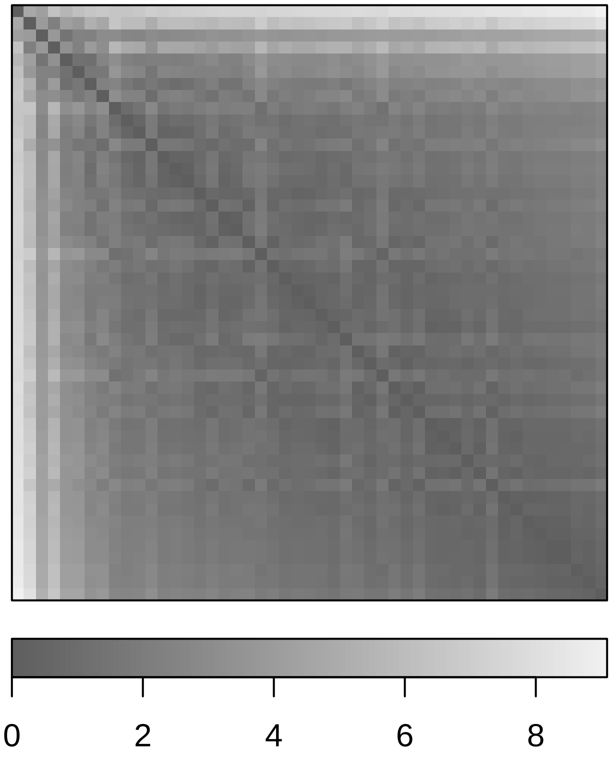
```r
ggplot(data = new_sldta) +
  geom_histogram(aes(x = expend), binwidth = 0.1) +
  labs(x="Expenditures per legislator in thousands",
       y="Frequency",
       title="Expenditures per Legislator") +
  theme_bw()
```



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.

From the ODI output, it is not visually clear if there is possible clustering in the data because the ODI does not present any obvious squares or patterns. Further, calculating the Hopkins statistics, which is less than 0.5 makes it hard to believe that there is clusterability in the given data.

```
# scale and calculate (and store) distance matrix
new_dist <- dist(new_sldta,
                 method = "euclidean")
# generate ODI
dissplot(new_dist)
```



```
#Hopkins statistics
x<-hopkins(new_sldta, n = nrow(new_sldta)-1)>0.5
x
```

```
##     H
## FALSE
```

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.

It seems that one cluster is way heavier than the other one with more data points in it. This may suggest that clustering may not be the best fit for these data because one cluster may be driven by a few outliers. Therefore, without outliers, these data eventually might have been one cluster.

```
library(ggplot2)
#install.packages("clValid")
library(clValid)
```

```
set.seed(4)
kmeans <- kmeans(new_sldta,
              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] "1" "2" "3" "4" ...
##  $ 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"
```

```
kmeans
```

```
## K-means clustering with 2 clusters of sizes 43, 6
##
## Cluster means:
##    t_slength    slength salary_real     expend
## 1 -0.2930275 -0.2932285  -0.2833616 -0.2047966
## 2  2.1000302  2.1014710   2.0307585  1.4677087
##
## Clustering vector:
##  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 50
##  1  1  1  1  1  1  2  1  1  2  1  1  2  1  1  1  1  1  1  1  1  1  1  1
##
## Within cluster sum of squares by cluster:
## [1] 48.36594 40.36173
##  (between_SS / total_SS =  53.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

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.

Gaussian mixture model gives a very similar results from the k-means that one cluster has significantly more data points than the other one (44 vs. 5).

```
#install.packages("mixtools")
library(mixtools) # fitting GMMs via EM (mixture model)
```

```
## mixtools package, version 1.1.0, Released 2017-03-10
## This package is based upon work supported by the National Science Foundation under Grant No. SES-0518
```

```
#install.packages("plotGMM")
library(plotGMM) # customizing GMM plot for visual output
gmm <- mvnormalmixEM(new_sldta,
                     k = 2)
```

```
## number of iterations= 28
```

```
cluster1<-gmm$posterior[,1]>0.5
cluster2<-gmm$posterior[,1]<0.5
summary(gmm)
```

```
## summary of mvnormalmixEM object:
##           comp 1    comp 2    comp 3     comp 4
## lambda  0.334039  0.665961  0.334039  0.6659610
## mu1    -0.300169 -0.357820 -0.549077 -0.1496554
## mu2     0.150562  0.179479  0.275411  0.0750655
## loglik at estimate:  -120.7838
```

```
sum(cluster1)
```

```
## [1] 18
```

```
sum(cluster2)
```

```
## [1] 31
```

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.

```
#install.packages("ppclust")
library(ppclust)
library(factoextra)
```

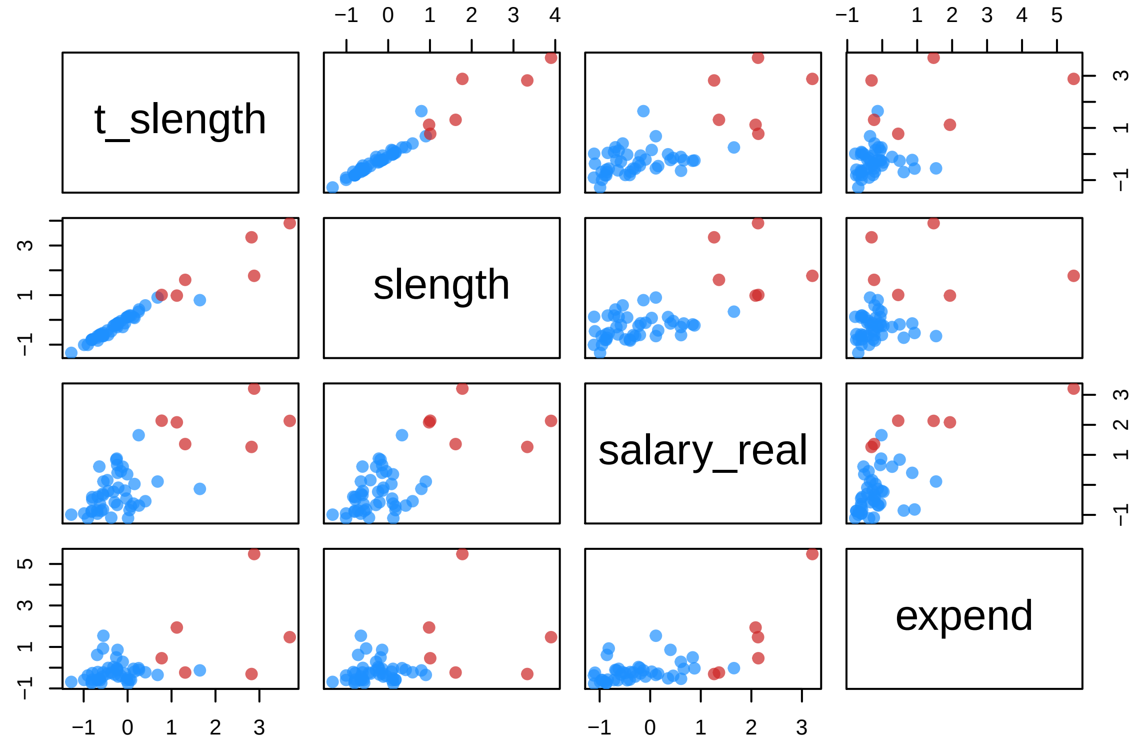
```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```
library(cluster)
library(fclust)
```

```
##
## Attaching package: 'fclust'
```

```
## The following object is masked from 'package:seriation':
##
##     VAT
```

```
res.fcm<-fcm(new_sldta, centers=2)
plotcluster(res.fcm, cp=1, trans=TRUE)
```



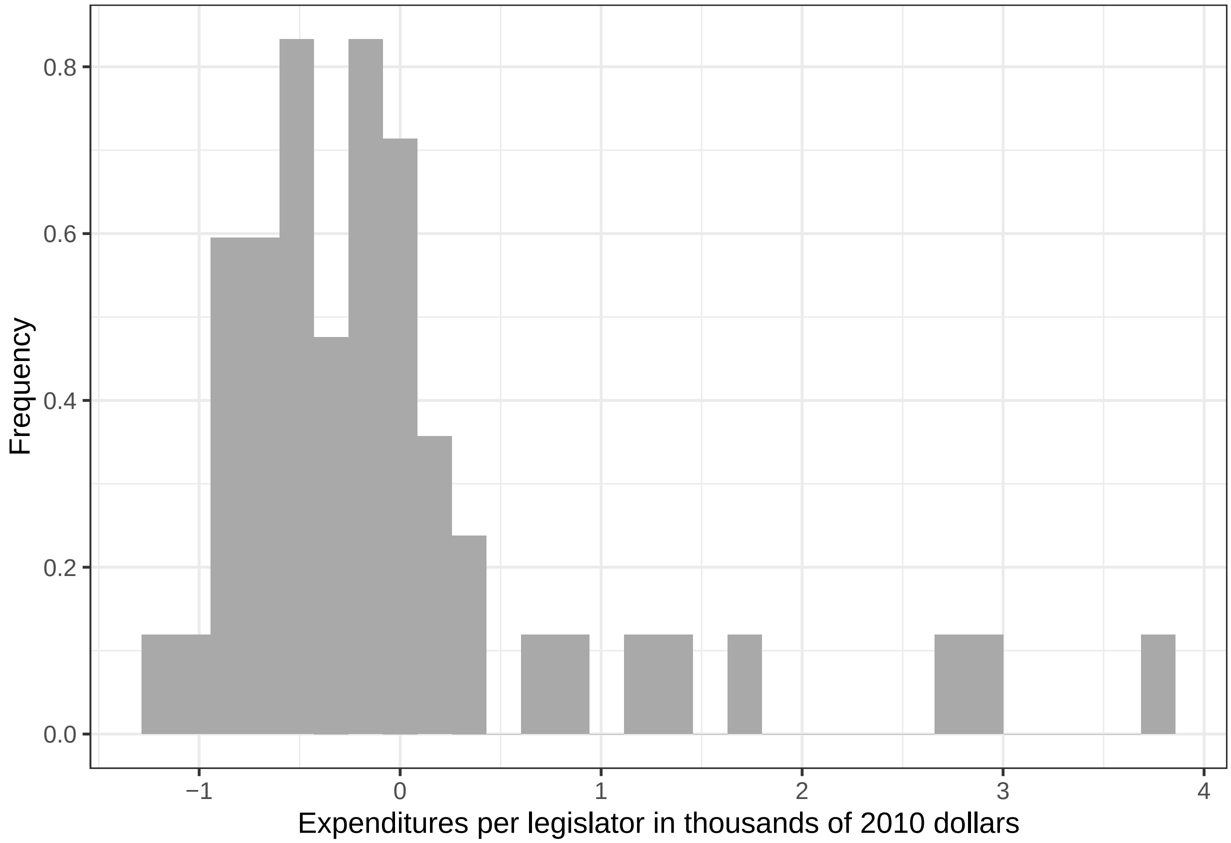
8. Compare output of all in a visually useful, simple way (e.g., plotting by state cluster assignment across two features like salary and expenditures).

Graphs of all the methods show very similar trends that one cluster include the most data points and the other cluster seems to consist of a few outliers. Also, looking at the fuzzy c-means, there seems to be correlation between salary and the length of session of legislation. However, some outliers might be driving this correlation.

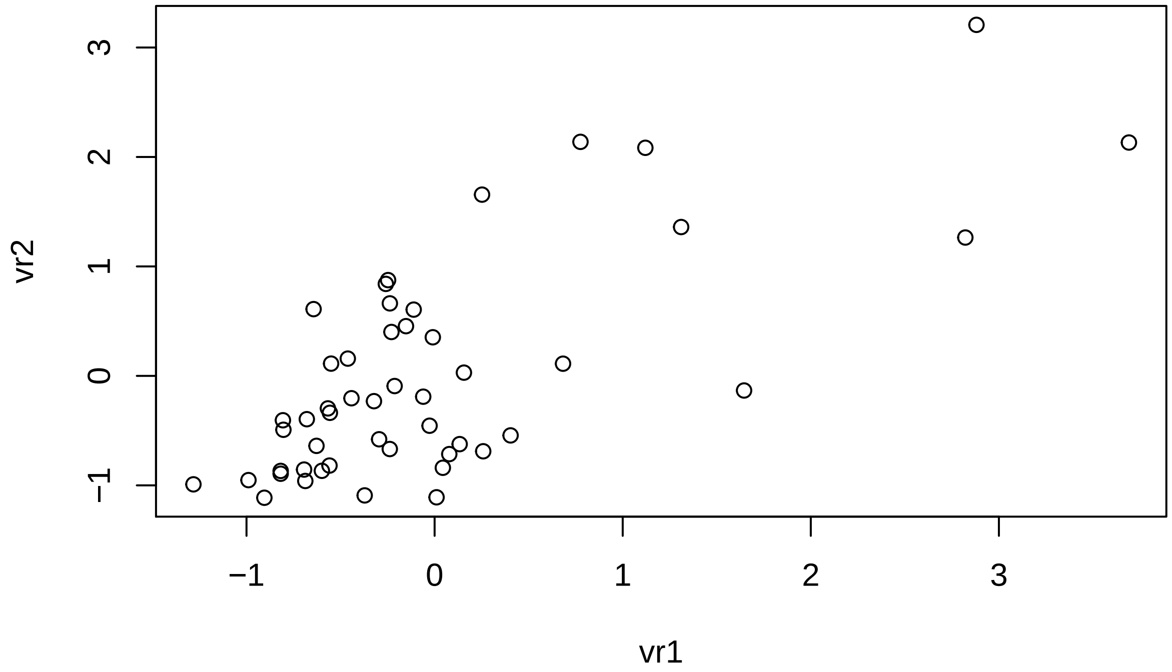
```
ggplot(data.frame(x = gmm$x[,1])) +
  geom_histogram(aes(x, ..density..), fill = "darkgray") +
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm$mu[1], gmm$sigma[[1]], lam = gmm$lambda[1]),
                colour = "darkblue") +
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm$mu[2], gmm$sigma[[2]], lam = gmm$lambda[2]),
                colour = "darkred") +
  xlab("Expenditures per legislator in thousands of 2010 dollars") +
  ylab("Frequency") +
  theme_bw()
```

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

```
## Warning: Computation failed in `stat_function()`:
## Non-numeric argument to mathematical function
## Warning: Computation failed in `stat_function()`:
## Non-numeric argument to mathematical function
```

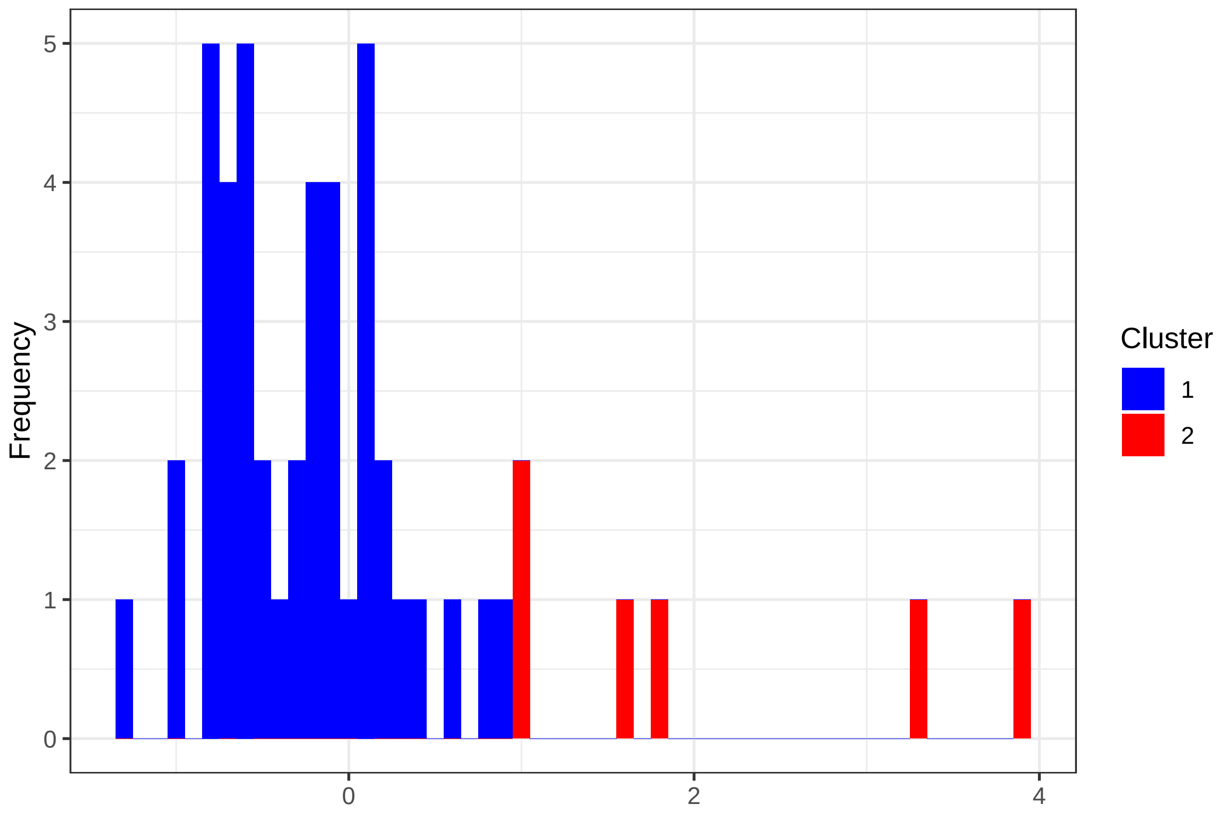


```
vr1<-gmm$x[,1]
vr2<-gmm$x[,3]
plot(vr1,vr2)
```



```
#vr1=total length(t_slength)
#vr2=salary (salary_real)

new_sldta$Cluster <- as.factor(kmeans$cluster)
t <- as.table(kmeans$cluster)
t <- data.frame(t)
rownames(t) <- state$State
colnames(t)[colnames(t)=="Freq"] <- "Assignment"
t$Var1 <- NULL
ggplot(new_sldta, aes(slength, fill = Cluster)) +
  geom_histogram(binwidth = 0.1) +
  theme_bw() +
  scale_fill_manual(values=c("blue", "red")) +
  labs(x = "",
       y = "Frequency")
```



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).

```r
#install.packages("mclust")
library(mclust)
```

```
## Package 'mclust' version 5.4.5
## Type 'citation("mclust")' for citing this R package in publications.

##
## Attaching package: 'mclust'

## The following object is masked from 'package:mixtools':
##
##     dmvnorm

## The following object is masked from 'package:purrr':
##
##     map
```

```r
#install.packages("cluster")
library(cluster)
library(clValid)
```

```
new_sldta<-select(new_sldta, -c(Cluster))
int <- as.matrix(new_sldta)
#Internal validity of K-means on all four continuous variables
k_int <- clValid(int, 2:10,
                    clMethods = c("kmeans"),
                    validation = "internal"); optimalScores(k_int)
```

```
##                  Score Method Clusters
## Connectivity 8.4460317 kmeans        2
## Dunn         0.2580885 kmeans        3
## Silhouette   0.6457584 kmeans        2
```

```
g_int <- clValid(int, 2:3,
                    clMethods = c("model"),
                    validation = "internal"); optimalScores(g_int)
```

```
##                   Score Method Clusters
## Connectivity 10.7392857  model        2
## Dunn          0.1522487  model        2
## Silhouette    0.6313660  model        2
```

```
f_int <- clValid(int, 2:4,
                    clMethods = c("fanny"),
                    validation = "internal"); optimalScores(f_int)
```

```
##                    Score Method Clusters
## Connectivity 17.61230159  fanny        2
## Dunn          0.05722842  fanny        4
## Silhouette    0.33819495  fanny        2
```

10. Discuss the validation output.

a. What can you take away from the fit?
b. Which approach is optimal? And optimal at what value of k?
c. What are reasons you could imagine selecting a technically "sub-optimal" partitioning method, regardless of the validation statistics?

It seems that having 2 clusters is generally optimal for all methods. Across all the methods, it can be shown that they favor 2 clusters looking at the silhouette scores.

It can be the case that we know something about the data that favor a certain method over the other. For example, some data may be very strict about how many categories they can belong to such as that one is either male or female. In other words, if the data already have some characteristics that define themselves, validation statistics may be a less important element to consider.