UML HW3

Yaoxi SHI

24 October 2019

1 Load Data

```
# load data
load("legprof-components.v1.0.RData")
```

2 Data Manipulation

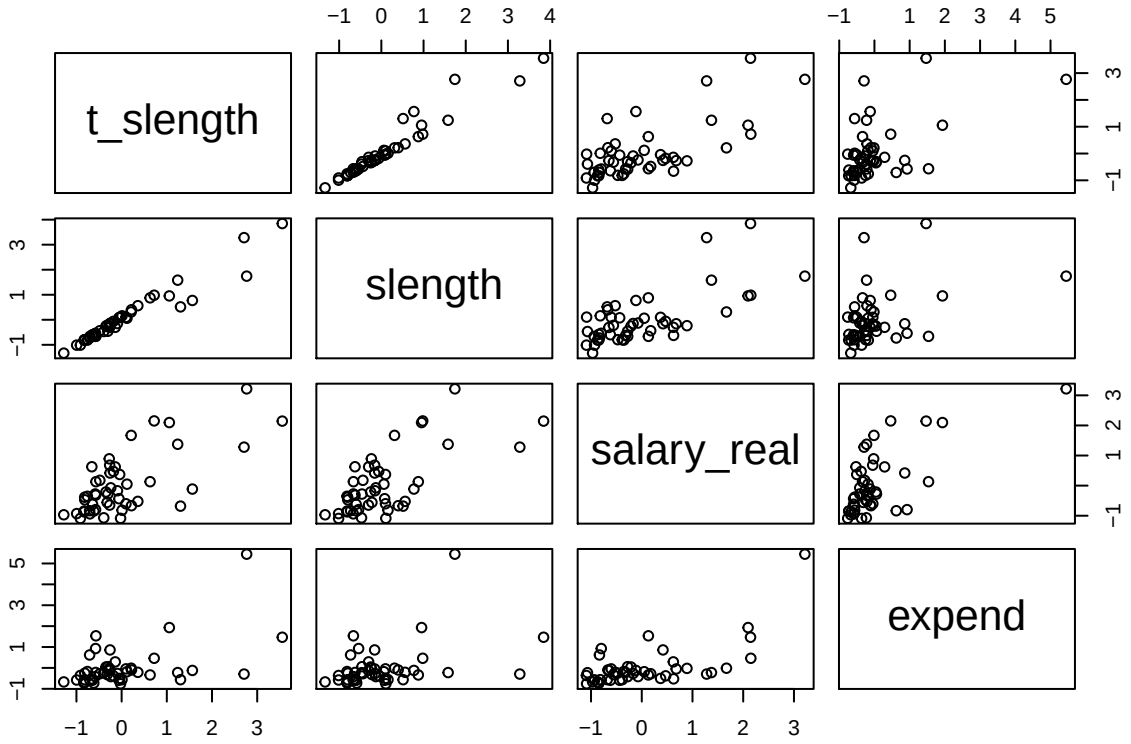
```
legprof=select(x, "stateabv", "t_slength", "slength", "salary_real", "expend") %>%
  as.data.frame() %>%
  filter(x$year==2009 | x$year==2010) %>%
  na.omit()
legprof_scaled=as.data.frame(scale(legprof[2:5]))
legprof_scaled$state=legprof$stateabv
```

3 EDA

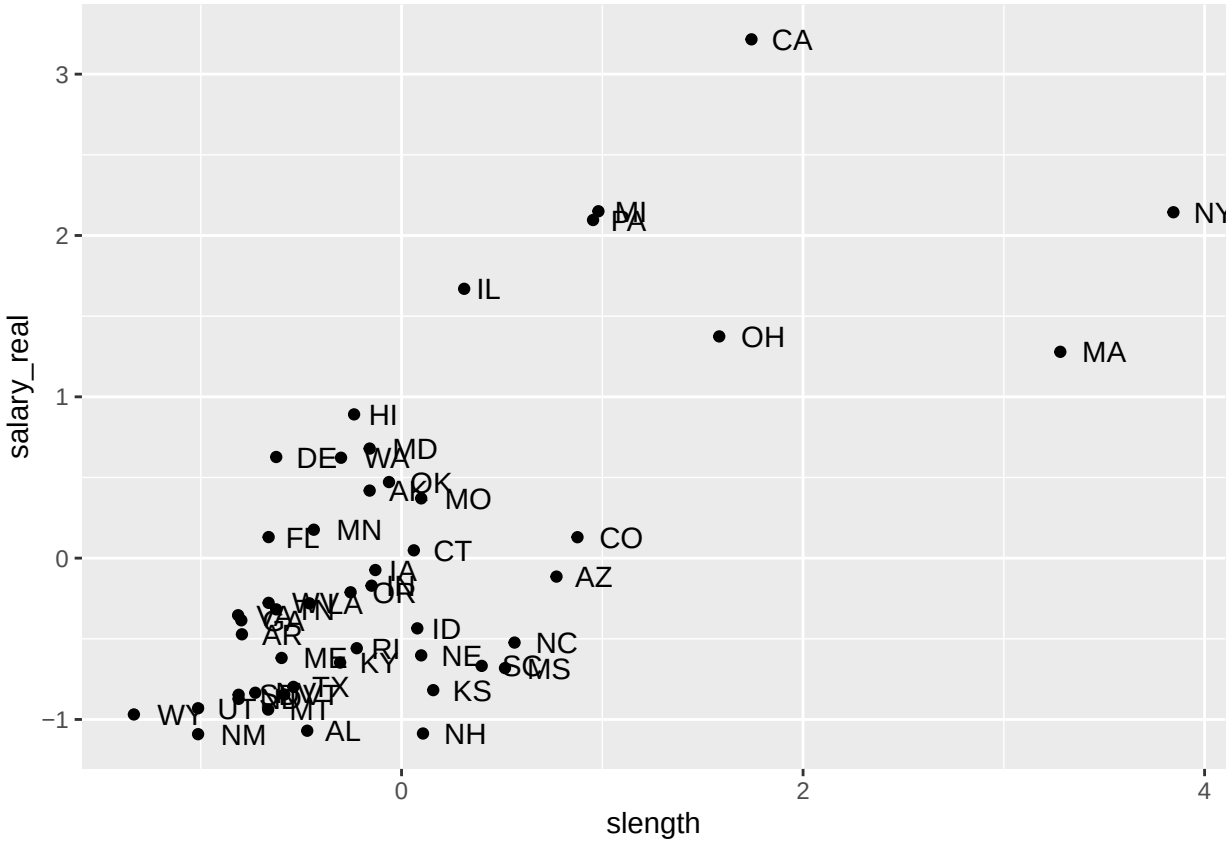
```
summary(legprof)
```

```
##    stateabv           t_slength         slength        salary_real
##  Length:48          Min.   : 40.00   Min.   : 40.00   Min.   :  0.00
##  Class :AsIs        1st Qu.: 96.81   1st Qu.: 92.29   1st Qu.: 18.85
##  Mode  :character   Median :127.78   Median :122.50   Median : 39.26
##                     Mean   :150.70   Mean   :139.68   Mean   : 54.06
##                     3rd Qu.:162.93   3rd Qu.:154.36   3rd Qu.: 75.46
##                     Max.   :458.15   Max.   :427.15   Max.   :213.41
##      expend
##  Min.   :  70.43
##  1st Qu.: 268.74
##  Median : 525.89
##  Mean   : 737.29
##  3rd Qu.: 719.32
##  Max.   :5523.10
```

```
pairs(legprof_scaled[1:4])
```



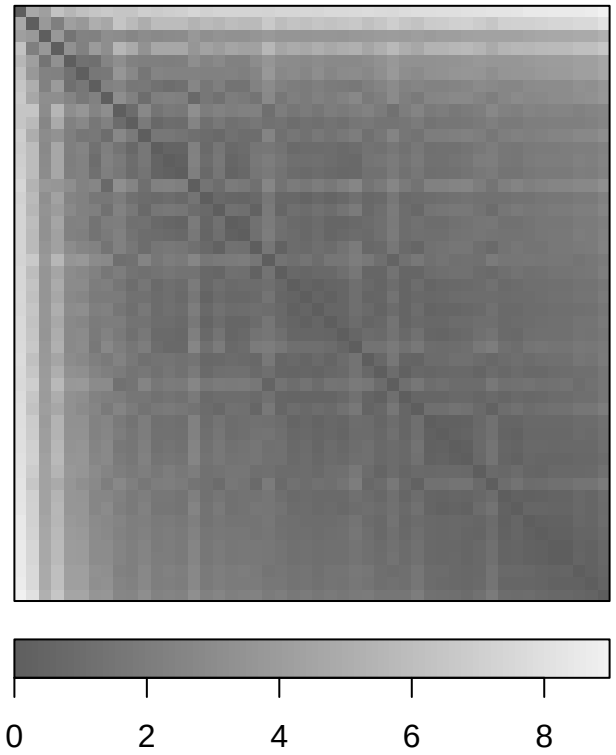

```
ggplot(legprof_scaled, aes(x=slength, y=salary_real))+
  geom_point()+
  geom_text(aes(label=state), hjust=-0.5, vjust=0.5)
```



We can roughly observe patterns in the session length versus salary plot, there is a group of states in the lower left corner, suggesting that they have lower salary and shorter session length, while another group of states locates in more upper right corner, with longer session length and higher salary, this group of state (such as CA, MI, PA, NY, MA, OH, IL) seem to be higher in legislative professionalism.

4 Diagnose clusterability

```
#ODI
legprof_dist=dist(legprof_scaled[,-5], method = "euclidean")
dissplot(legprof_dist)
```



From the ODI plot, though the clustering pattern is not very obvious, it's still different from the plot that is generated from the random data, we can find that there are roughly 2 or 3 darker blocks exist in this plot, suggesting that there is non-random structure in the data that could potentially be clustered into groups.

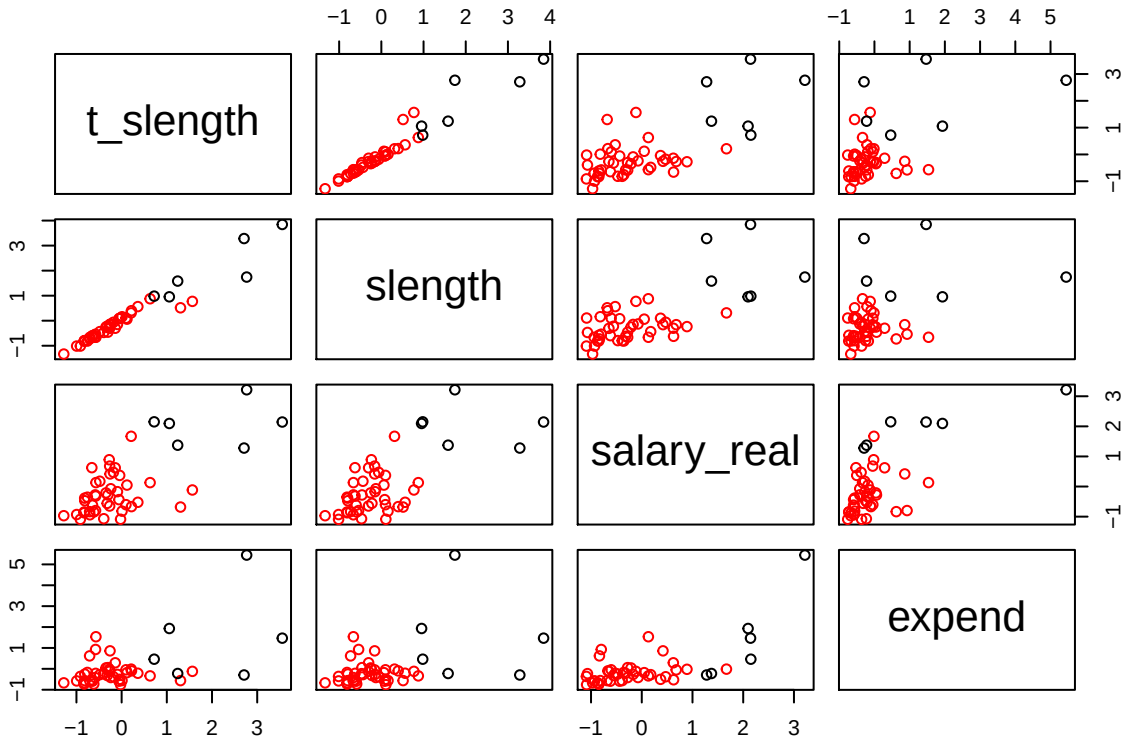
5 K-means

```
legpro_kmeans=kmeans(legprof_scaled[,-5],
        center=2,
        nstart=15)
str(legpro_kmeans)
```

```
## List of 9
##  $ cluster     : Named int [1:48] 2 2 2 2 1 2 2 2 2 2 ...
##   ..- attr(*, "names")= chr [1:48] "1" "2" "3" "4" ...
##  $ centers     : num [1:2, 1:4] 2.008 -0.287 2.064 -0.295 2.043 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:4] "t_slength" "slength" "salary_real" "expend"
##  $ totss       : num 188
##  $ withinss    : num [1:2] 39.5 48.3
##  $ tot.withinss: num 87.8
##  $ betweenss   : num 100
##  $ size        : int [1:2] 6 42
```

```
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

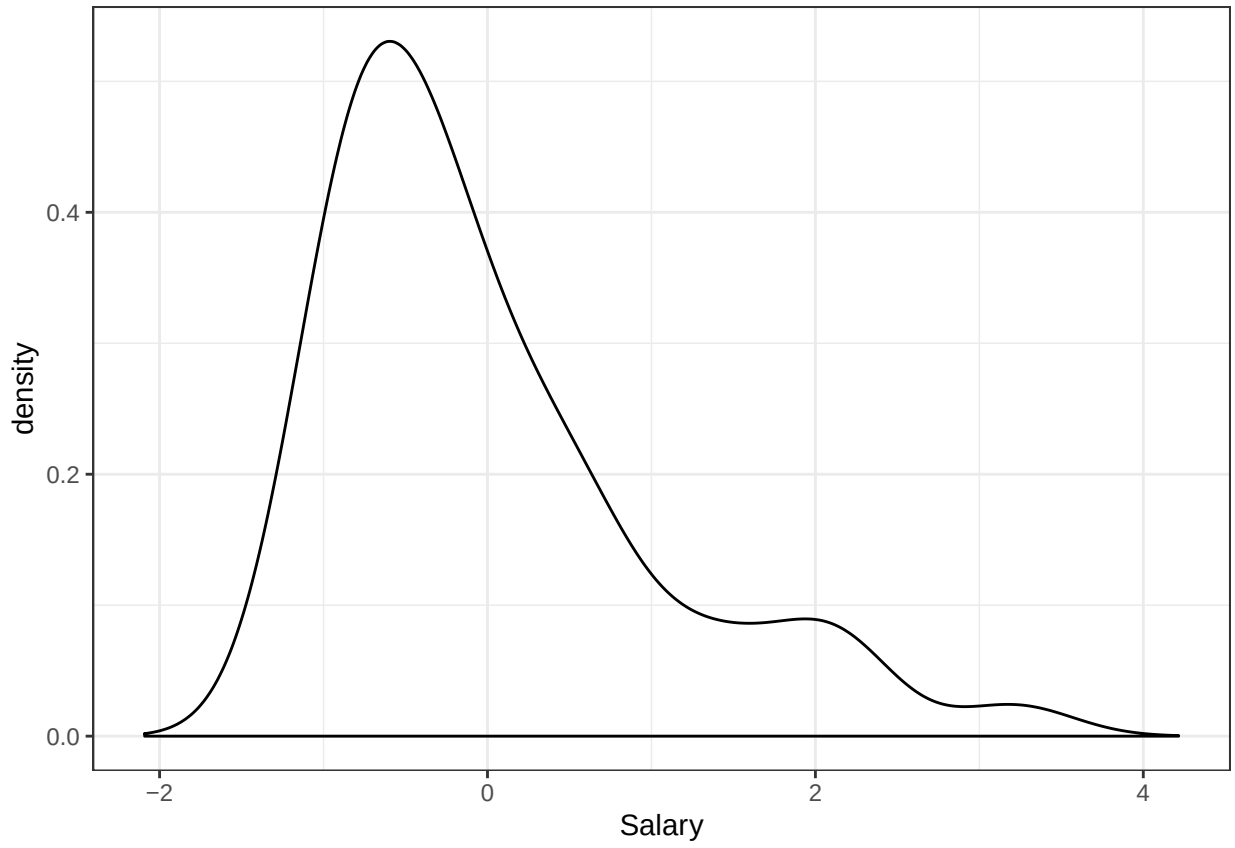
```
legprof_scaled$kmeans=as.factor(legpro_kmeans$cluster)

with(legprof_scaled, pairs(legprof_scaled[1:4], col=c(1:2)[legpro_kmeans$cluster]))
```



The states are clustered into two groups, one group contains 42 states while the other only contains 6 states, the clustering results is almost the same as the grouping pattern we have obeserved in the scatter plot in EDA.

6 EM algorithm

```
ggplot(legprof_scaled, aes(x = salary_real)) +
  geom_density()+
  xlim(min(legprof_scaled$salary_real)-1, max(legprof_scaled$salary_real)+1) +
  theme_bw() +
  labs(x = "Salary")
```




```
set.seed(111)
legprof_gmm=normalmixEM(legprof_scaled$salary_real, k = 2)
```

```
## number of iterations= 58
```

```
summary(legprof_gmm)
```

```
## summary of normalmixEM object:
##           comp 1   comp 2
## lambda  0.590997 0.409003
## mu     -0.582349 0.841471
## sigma   0.342395 1.013174
## loglik at estimate:  -56.74103
```

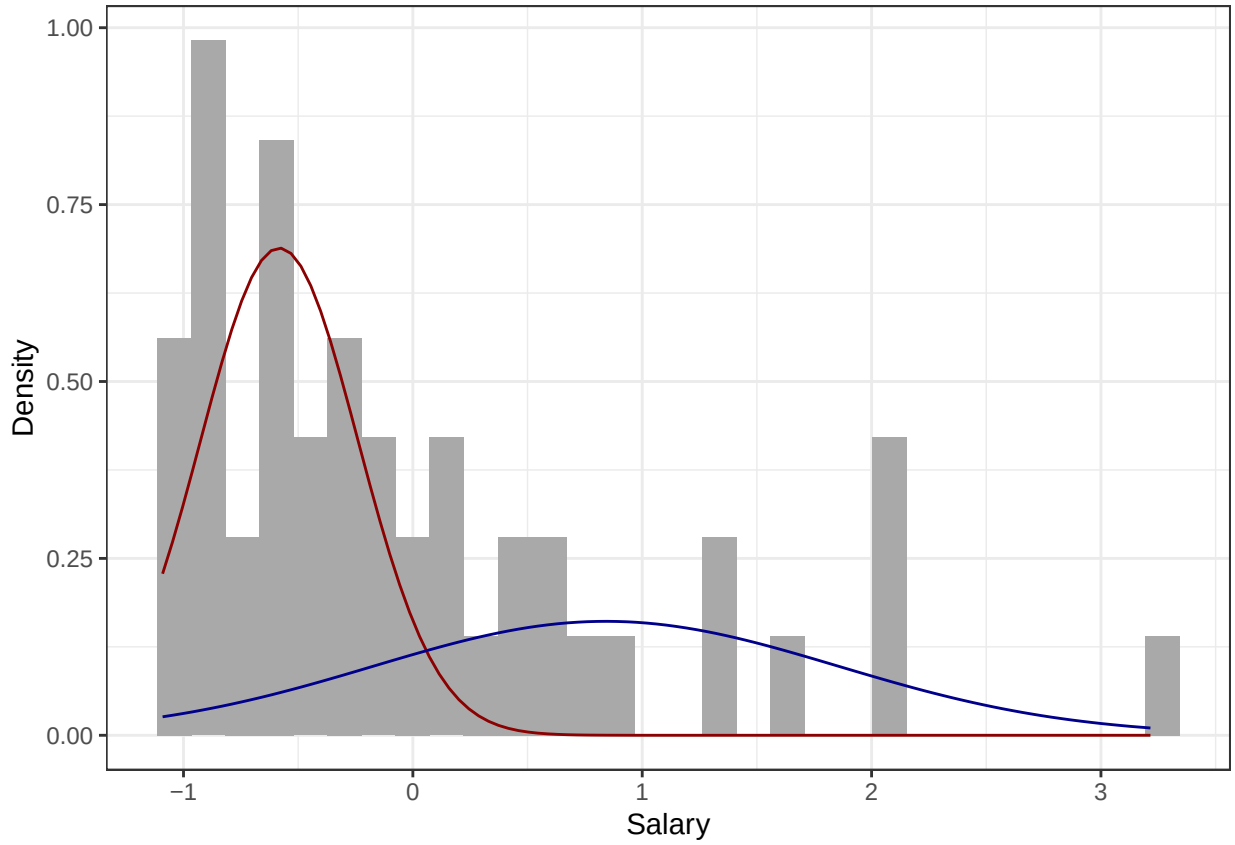
```
str(legprof_gmm)
```

```
## List of 9
##  $ x         : num [1:48] -1.07 0.419 -0.114 -0.472 3.216 ...
##  $ lambda    : num [1:2] 0.591 0.409
##  $ mu        : num [1:2] -0.582 0.841
##  $ sigma     : num [1:2] 0.342 1.013
##  $ loglik    : num -56.7
##  $ posterior : num [1:48, 1:2] 9.02e-01 6.10e-02 7.24e-01 9.04e-01 1.27e-25 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:2] "comp.1" "comp.2"
##  $ all.loglik: num [1:59] -84.8 -67.3 -66.6 -65 -62 ...
##  $ restarts  : num 0
```

```
##  $ ft        : chr "normalmixEM"
##  - attr(*, "class")= chr "mixEM"
```

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

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




```
em_posterior <- data.frame(cbind(legprof_gmm$x, legprof_gmm$posterior))

rownames(em_posterior) <- legprof_scaled$state
em_posterior$component <- ifelse(em_posterior$comp.1 > 0.3, 2, 1)
table(em_posterior$component)
```

```
##
##  1  2
## 14 34
```

```
em_posterior
```

```
##             V1       comp.1     comp.2 component
## AL -1.06990427 9.019098e-01 0.09809023         2
## AK  0.41862783 6.103939e-02 0.93896061         1
## AZ -0.11442305 7.239611e-01 0.27603890         2
## AR -0.47214169 9.039293e-01 0.09607070         2
## CA  3.21583786 1.266305e-25 1.00000000         1
## CO  0.12974715 3.862775e-01 0.61372249         2
## CT  0.04835710 5.156930e-01 0.48430696         2
## DE  0.62701413 8.472611e-03 0.99152739         1
## FL  0.13052439 3.850307e-01 0.61496925         2
## GA -0.38537057 8.829444e-01 0.11705562         2
## HI  0.89106985 4.075512e-04 0.99959245         1
## ID -0.43526270 8.961041e-01 0.10389592         2
## IL  1.66948443 2.419459e-09 1.00000000         1
## IN -0.17072625 7.737095e-01 0.22629049         2
## IA -0.07372802 6.808356e-01 0.31916437         2
## KS -0.81822147 9.280685e-01 0.07193153         2
## KY -0.64742213 9.251642e-01 0.07483582         2
## LA -0.28001365 8.423394e-01 0.15766056         2
## ME -0.61866626 9.231406e-01 0.07685940         2
## MD  0.67913011 4.863241e-03 0.99513676         1
## MA  1.27885888 1.799274e-06 0.99999820         1
## MI  2.15025554 1.453241e-13 1.00000000         1
## MN  0.17617615 3.131706e-01 0.68682943         2
## MS -0.68157293 9.269561e-01 0.07304390         2
## MO  0.37045830 9.022482e-02 0.90977518         1
## MT -0.93853610 9.209379e-01 0.07906208         2
## NE -0.60276345 9.218069e-01 0.07819310         2
## NV -0.83436517 9.275753e-01 0.07242469         2
## NH -1.08703436 8.982723e-01 0.10172773         2
## NM -1.09110386 8.973607e-01 0.10263929         2
## NY  2.14415128 1.662224e-13 1.00000000         1
## NC -0.52336744 9.125728e-01 0.08742723         2
## ND -0.87233232 9.258723e-01 0.07412774         2
## OH  1.37436411 3.974752e-07 0.99999960         1
## OK  0.47158547 3.850020e-02 0.96149980         1
## OR -0.21160279 8.032755e-01 0.19672454         2
## PA  2.09591378 4.758852e-13 1.00000000         1
## RI -0.55844657 9.172155e-01 0.08278452         2
## SC -0.66787550 9.263145e-01 0.07368554         2
## SD -0.84693366 9.270971e-01 0.07290288         2
## TN -0.31753194 8.591401e-01 0.14085991         2
## TX -0.79809962 9.284960e-01 0.07150401         2
## UT -0.93045963 9.216837e-01 0.07831629         2
## VT -0.84588214 9.271403e-01 0.07285967         2
## VA -0.35394815 8.728651e-01 0.12713494         2
## WA  0.62240128 8.889641e-03 0.99111036         1
## WV -0.27720319 8.409603e-01 0.15903971         2
## WY -0.96901876 9.177297e-01 0.08227033         2
```

```
legprof_scaled$em=as.factor(em_posterior$component)
```

There are two distributions of the salary of legislators in different states, one with lower mean suggesting lower level of legislative professionalism sincelegislative professionalism is highly related with legislator compensation, another distribution has a higher mean, suggesting higher level of legislative professionalism.

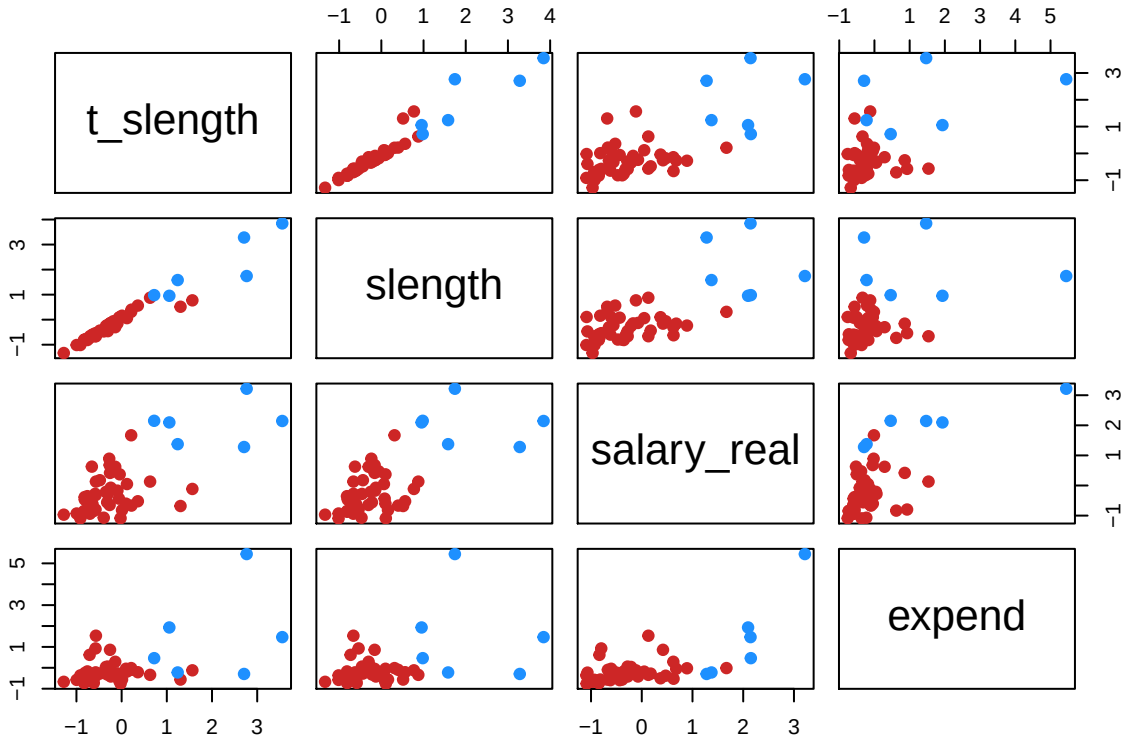
7 Fuzzy C-means

```
leg_fcm=fcm(legprof_scaled[1:4], center=2)
summary(leg_fcm)
```

```
## Summary for 'leg_fcm'
##
## Number of data objects:  48
##
## Number of clusters:  2
##
## Crisp clustering vector:
##  [1] 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 1 2 2 1 2
## [36] 2 1 2 2 2 2 2 2 2 2 2 2 2
##
## Initial cluster prototypes:
##            t_slength    slength salary_real     expend
## Cluster 1  0.7203749  0.9806336   2.1502555  0.4616150
## Cluster 2 -1.2798768 -1.3331515  -0.9690188 -0.6714222
##
## Final cluster prototypes:
##            t_slength    slength salary_real     expend
## Cluster 1  1.8119173  1.8520733   1.7811444  1.1752025
## Cluster 2 -0.3372842 -0.3311178  -0.3277574 -0.2185666
##
## Distance between the final cluster prototypes
##           Cluster 1
## Cluster 2  15.77545
##
## Difference between the initial and final cluster prototypes
##           t_slength   slength salary_real    expend
## Cluster 1 1.0915424 0.8714396  -0.3691111 0.7135875
## Cluster 2 0.9425927 1.0020337   0.6412614 0.4528556
##
## Root Mean Squared Deviations (RMSD): 1.597681
## Mean Absolute Deviation (MAD): 12.16885
##
## Membership degrees matrix (top and bottom 5 rows):
##    Cluster 1 Cluster 2
## 1 0.02738938 0.9726106
## 2 0.14608804 0.8539120
## 3 0.42948278 0.5705172
## 4 0.02165084 0.9783492
## 5 0.73410396 0.2658960
## ...
##     Cluster 1 Cluster 2
## 45 0.02898264 0.9710174
## 46 0.02011605 0.9798840
## 47 0.10182414 0.8981759
```

```
## 48 0.01117682 0.9888232
## 50 0.07559702 0.9244030
##
## Descriptive statistics for the membership degrees by clusters
##           Size       Min        Q1      Mean    Median        Q3       Max
## Cluster 1    6 0.7341040 0.7827334 0.8085230 0.8050030 0.8452859 0.8730042
## Cluster 2   42 0.5705172 0.9191211 0.9226321 0.9563491 0.9795003 0.9954766
##
## Dunn's Fuzziness Coefficients:
## dunn_coeff normalized
##  0.8537694  0.7075388
##
## Within cluster sum of squares by cluster:
##        1        2
## 39.45254 48.33873
## (between_SS / total_SS =  48.54%)
##
## Available components:
##  [1] "u"           "v"           "v0"          "d"           "x"
##  [6] "cluster"     "csize"       "sumsqrs"     "k"           "m"
## [11] "iter"        "best.start" "func.val"    "comp.time"   "inpargs"
## [16] "algorithm"   "call"
```

```
ppclust::plotcluster(leg_fcm)
```

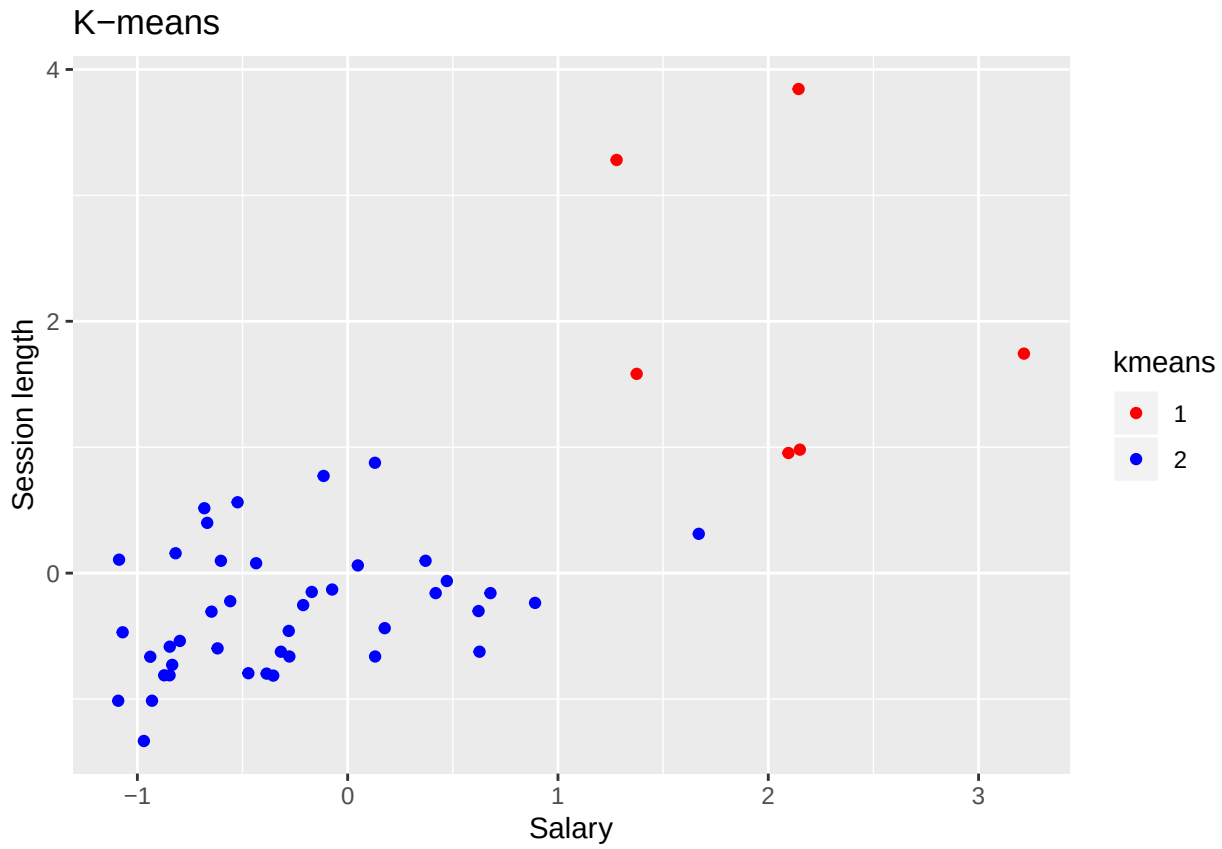


```
legprof_scaled$fcm=as.factor(leg_fcm$cluster)
```

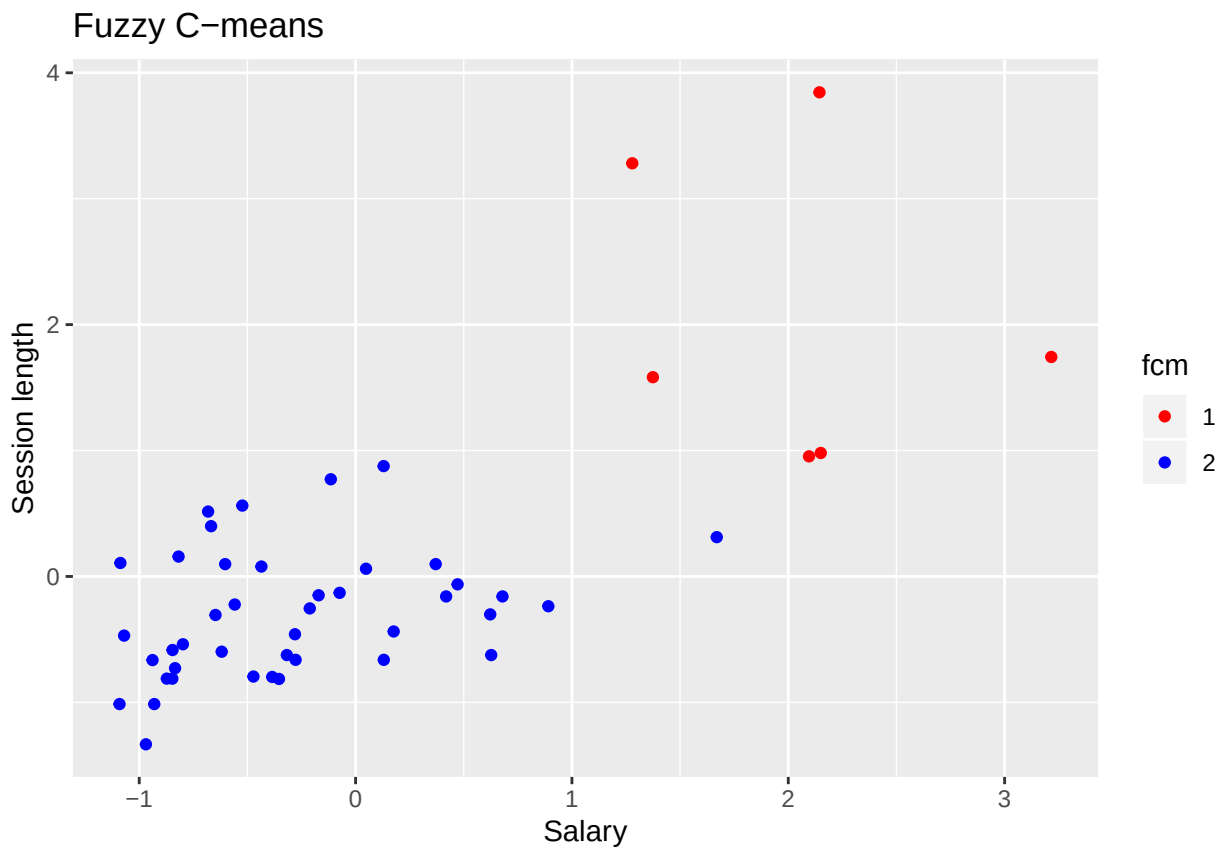
The results of fuzzy C-means clustering is the same as the results from k means clustering, one group with higher session length, higher salary and higher expenditures, suggesting these states have higher level of legislative professionalism, another group are lower in there three dimensions, we could interprete them as lower in legislative professionalism.

8 Visulization

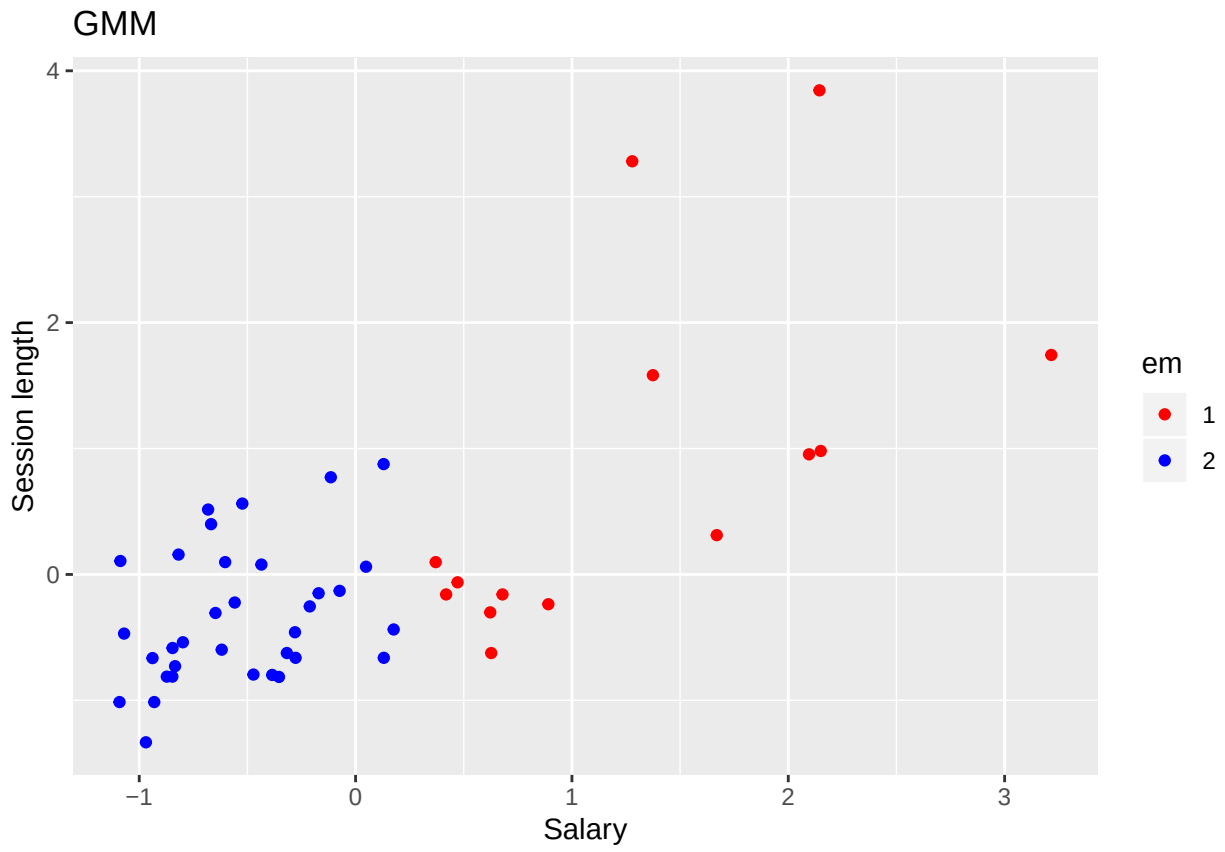
```
ggplot(legprof_scaled, aes(x=salary_real, y=slength, color=kmeans))+
  geom_point()+
  scale_color_manual(values=c("Red","Blue"))+
  labs(x="Salary", y="Session length", title = "K-means")
```




```
ggplot(legprof_scaled, aes(x=salary_real, y=slength, color=fcm))+
  geom_point()+
  scale_color_manual(values=c("Red","Blue"))+
  labs(x="Salary", y="Session length", title = "Fuzzy C-means")
```




```
ggplot(legprof_scaled, aes(x=salary_real, y=slength, color=em))+
  geom_point()+
  scale_color_manual(values=c("Red","Blue"))+
  labs(x="Salary", y="Session length", title = "GMM")
```



9 Validation: Average Silhouette Width

```
# kmeans
leg_kmeans_int = as.matrix(legprof_scaled[1:4])

kmeans_internal = clValid(leg_kmeans_int, 2:10,
                    clMethods = c("kmeans"),
                    validation = "internal");
summary(kmeans_internal)
```

```
##
## Clustering Methods:
##  kmeans
##
## Cluster sizes:
##  2 3 4 5 6 7 8 9 10
##
## Validation Measures:
##                             2       3       4       5       6       7       8       9      10
##
## kmeans Connectivity   8.5683 11.0183 18.1651 20.1651 23.6810 25.8476 36.4726 44.8750 45.4024
##        Dunn           0.1726  0.2597  0.2456  0.2456  0.1214  0.1214  0.1871  0.1846  0.2515
##        Silhouette     0.6390  0.6054  0.4824  0.4611  0.3328  0.3210  0.3169  0.2854  0.3249
##
## Optimal Scores:
##
##              Score  Method Clusters
```

```
## Connectivity 8.5683 kmeans 2
## Dunn         0.2597 kmeans 3
## Silhouette   0.6390 kmeans 2
```

```
#GMM
em_internal = clValid(leg_kmeans_int, 2:10,
                      clMethods = c("model"),
                      validation = "internal");
```

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

```
summary(em_internal)
```

```
##
## Clustering Methods:
##  model
##
## Cluster sizes:
##  2 3 4 5 6 7 8 9 10
##
## Validation Measures:
##                            2       3       4       5       6       7       8       9      10
##
## model Connectivity  18.7095 23.7964 33.3683 60.1651 69.0651 54.4433 51.8206 63.7619 62.1766
##       Dunn           0.0833  0.0855  0.0554  0.0280  0.0391  0.0532  0.0935  0.0879  0.0928
##       Silhouette     0.4230  0.3854  0.2157  0.0962  0.0473  0.1822  0.2957  0.2091  0.2132
##
## Optimal Scores:
##
##              Score   Method Clusters
## Connectivity 18.7095 model  2
## Dunn          0.0935 model  8
## Silhouette    0.4230 model  2
```

```
#fcm
fcm_internal = clValid(leg_kmeans_int, 2:4,
                      clMethods = c("fanny"),
                      validation = "internal");
summary(fcm_internal)
```

```
##
## Clustering Methods:
##  fanny
##
## Cluster sizes:
##  2 3 4
##
## Validation Measures:
##                            2       3       4
##
## fanny Connectivity  18.4119 27.3401 38.5802
##       Dunn           0.0453  0.0457  0.0425
##       Silhouette     0.3324  0.2341  0.1978
##
## Optimal Scores:
##
```

```
##              Score   Method Clusters
## Connectivity 18.4119 fanny  2
## Dunn          0.0457 fanny  3
## Silhouette    0.3324 fanny  2
```

Average silhouette width was used as an indicator to validate the models, according to the results, k-means clustering has the highest average silhouette width, thus fit best to clustering the data becuase it has the largesr average silhouette width.

10

According to the validation results, if we only look at at Silhouette width, all three algorithms suggest that 2 clusters is the best fit for the data, but different validation methods have different optimal number of clusters. Also, different validation methods have different results of validating the methods, for example, silhouette width and dunn index suggests that gmm is better than fuzzy c means, while connectivity suggests the opposite.

For this dataset, k-means clustering is the optimal algorithm, it has obviously the largest silhouette width and dunn index, smallest connectivity, and the optimal value of k is 2.

I would select fuzzy c-means clustering method as sub-optimal one because in this case, hard partitioning works better than soft partitioning. Also, gmm only used salary as clustering indicator while fuzzy s-means take all the four features into consideration.