PS3 Submission_Sharon Shen

Sharon Shen

October 23, 2019

Problem Set 3 - Clustering via Partitioning

Load the state legislative professionalism data:

```
load("C:/Users/sharo/Box Sync/UChicago Courses/Unsupervised Machine Learning/PS3/State Leg Prof Data & Codebook/legprof-components.v1.0.RData")
summary(x)
```

```
##       fips         stateabv            state               sessid     
##  Min.   : 1.00   Length:950         Length:950         1973/4 : 50  
##  1st Qu.:17.00   Class :AsIs        Class :AsIs        1975/6 : 50  
##  Median :29.50   Mode  :character   Mode  :character   1977/8 : 50  
##  Mean   :29.32                                         1979/80: 50  
##  3rd Qu.:42.00                                         1981/2 : 50  
##  Max.   :56.00                                         1983/4 : 50  
##                                                        (Other):650  
##    t_slength        slength       salary_real         expend       
##  Min.   : 36.0   Min.   : 36.0   Min.   :  0.00   Min.   :  40.13  
##  1st Qu.: 91.0   1st Qu.: 85.2   1st Qu.: 20.11   1st Qu.: 219.93  
##  Median :128.5   Median :120.0   Median : 41.96   Median : 395.10  
##  Mean   :147.6   Mean   :136.4   Mean   : 55.82   Mean   : 599.51  
##  3rd Qu.:171.0   3rd Qu.:158.0   3rd Qu.: 80.08   3rd Qu.: 650.29  
##  Max.   :549.5   Max.   :521.9   Max.   :254.94   Max.   :5523.10  
##  NA's   :61      NA's   :61      NA's   :5        NA's   :5        
##       year           mds1              mds2         
##  Min.   :1974   Min.   :-1.8505   Min.   :-3.13760  
##  1st Qu.:1982   1st Qu.:-0.9257   1st Qu.:-0.34346  
##  Median :1992   Median :-0.3117   Median : 0.09395  
##  Mean   :1992   Mean   : 0.0000   Mean   : 0.00000  
##  3rd Qu.:2002   3rd Qu.: 0.4137   3rd Qu.: 0.30433  
##  Max.   :2011   Max.   : 8.5646   Max.   : 3.35495  
##                 NA's   :61        NA's   :61
```

Data wrangling

- Only include continuous variables
- Restrict to 2009/10
- Omit missing value
- Store state names as separate object

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------
------------------------------------- tidyverse 1.2.1 --
```

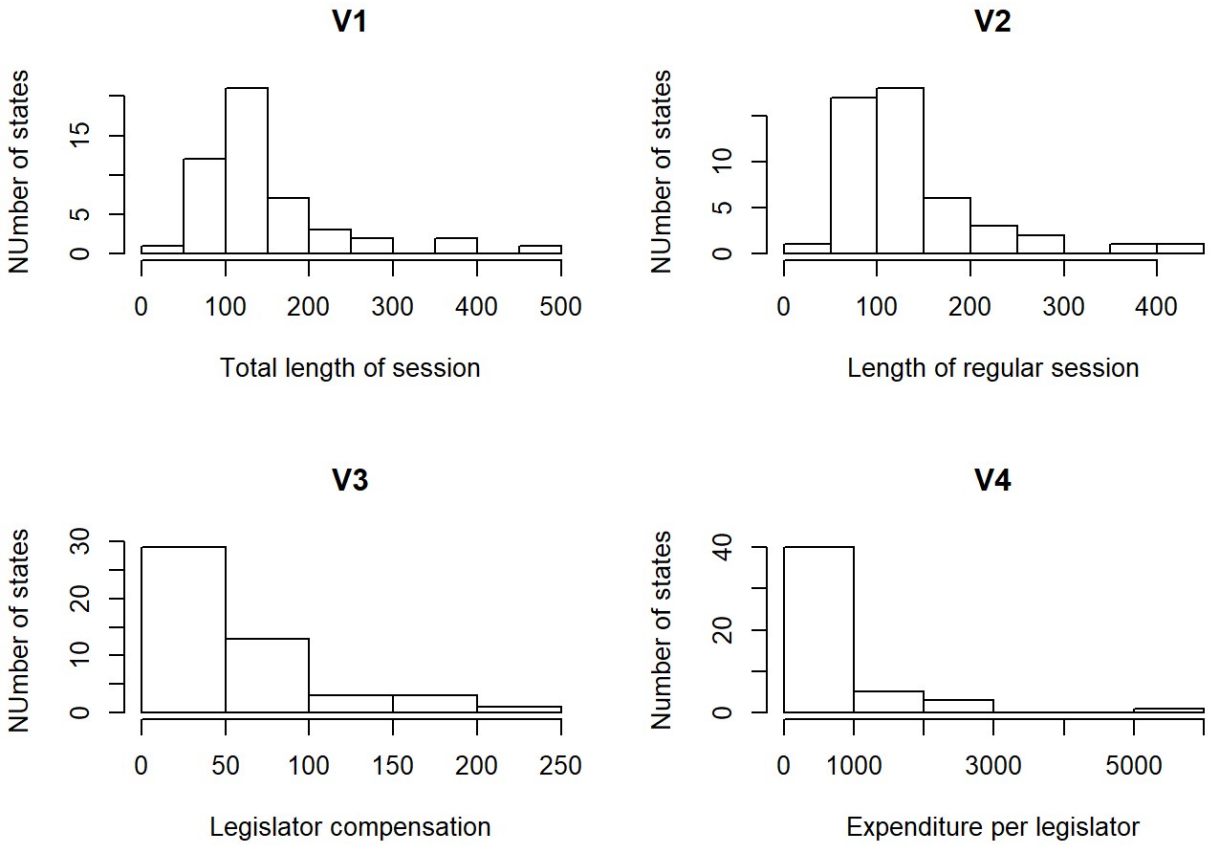
```
## v ggplot2 3.0.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.6
## v tidyr   0.8.1     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

```
## -- Conflicts ------------------------------------------------------------------
-------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

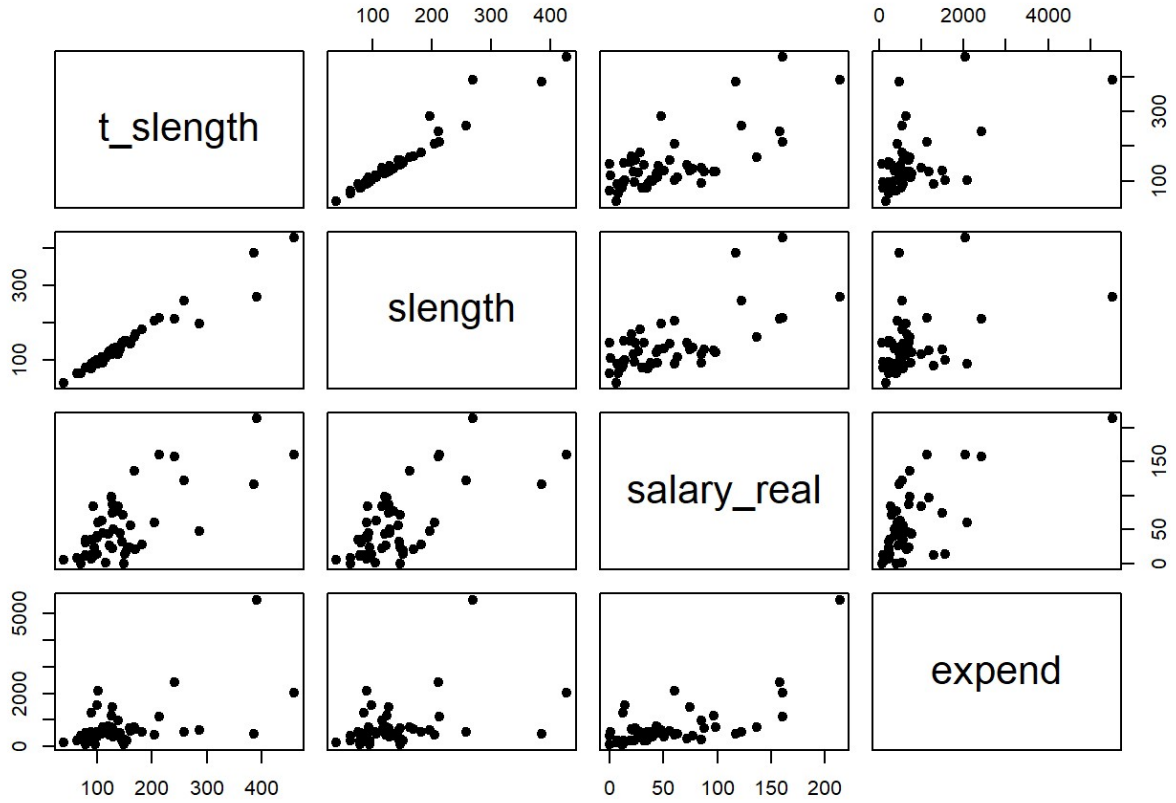
```
View(x)
x_sub <- x %>%
  dplyr::filter(sessid == "2009/10") %>%
  select(t_slength, slength, salary_real, expend) %>%
  drop_na()

View(x_sub)

# Have subset of statae variable to be used later for plotting
state <- x %>%
  dplyr::filter(sessid == "2009/10") %>%
  select(state,t_slength, slength, salary_real, expend) %>%
  drop_na()


# Check what happened with wisconsin
#subwisconsin <- x %>%
#  dplyr::filter(sessid == "2009/10" & state == "Wisconsin")
#View(subwisconsin)
```

Perform EDA

```
par(mfrow = c(2, 2)) # places all histograms in a single 2x2 plot pane

# Univariate analysis

hist(x_sub$t_slength,
     xlab = "Total length of session",
     ylab = "NUmber of states",
     main = "V1")
hist(x_sub$slength,
     xlab = "Length of regular session",
     ylab = "NUmber of states",
     main = "V2")
hist(x_sub$salary_real,
     xlab = "Legislator compensation",
     ylab = "NUmber of states",
     main = "V3")
hist(x_sub$expend,
     xlab = "Expenditure per legislator",
     ylab = "Number of states",
     main = "V4")
```



```
# Bivariate analysis

pairs(x_sub[,1:4], pch = 19)
```



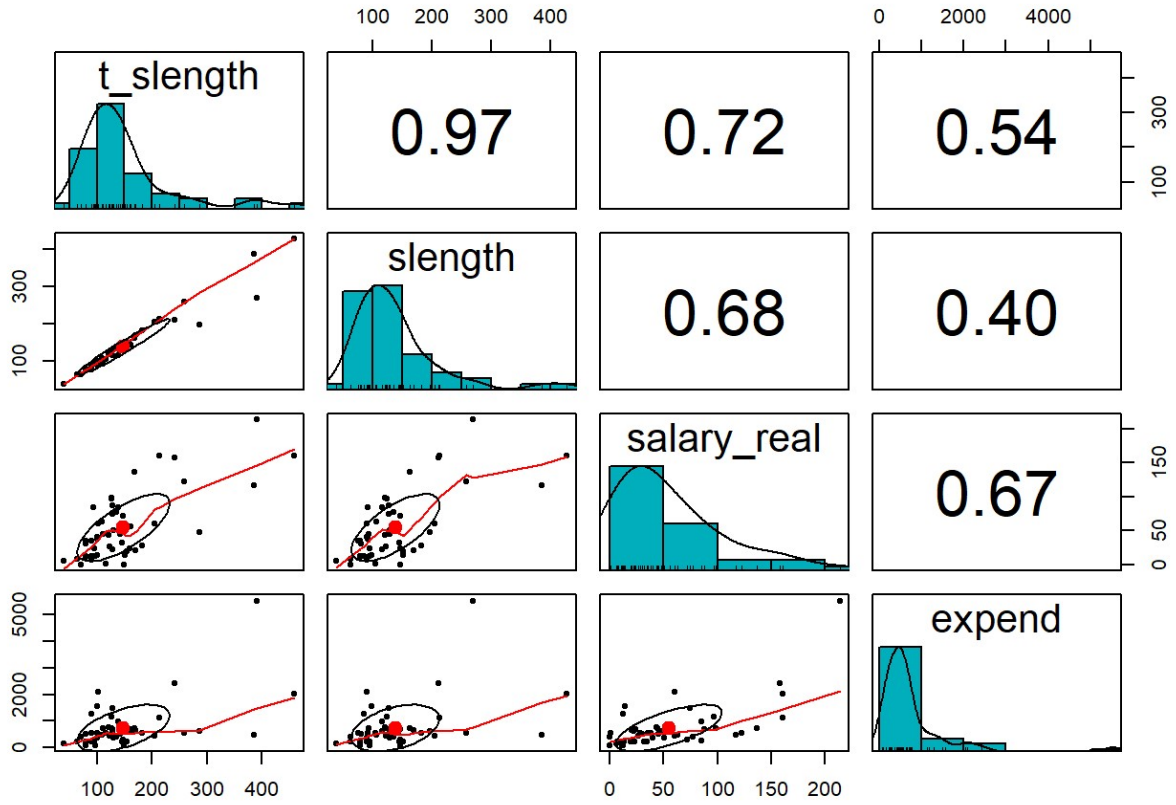

```
library(psych)
```

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

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
pairs.panels(x_sub[,-5],
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = TRUE # show correlation ellipses
             )
```



Diagnose clusterability

1. Informally

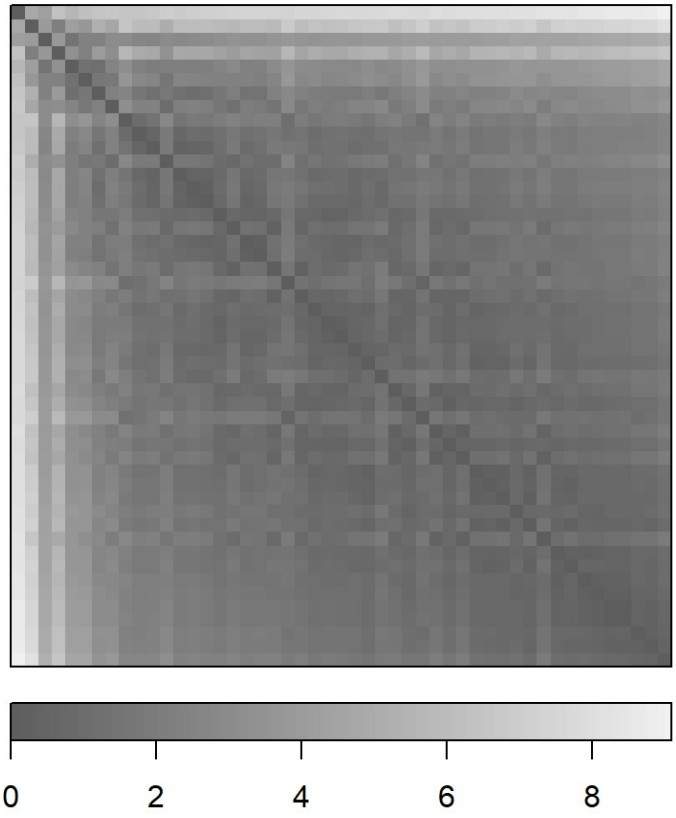
As the matrix scatterplot shown above, most observations seem to be concentrated in the bottom left hand corner, while a few at the other extreme.

2. Using ODI

```
library(seriation)
```

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

```
xsub_scaled <- scale(x_sub)
x_dist <- dist(xsub_scaled,
               method = "euclidean")


# generate ODI plot
dissplot(x_dist)
```



There seems to be weak grouping as the distinction between dark and light blocks is blurry on the diagonal line.

3. Using sparse sampling

```
# Calculate hopkins statistics
library(clustertend)
set.seed(123)
hopkins(xsub_scaled, n = nrow(xsub_scaled)-1)
```

```
## $H
## [1] 0.1723366
```

Based on the Hopkins statistics, there may not be meaningful clustering. Since the test statistics is 0.17, which is smaller than 0.5, it means we fail to reject null hypothesis, suggesting data could be randomly distributed.

Fit k-mean algorithm

```
library(ggplot2)
library(clValid)
```

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

```
## Loading required package: cluster
```

```
set.seed(123)

kmeans <- kmeans(xsub_scaled[,4],
              centers = 2,
              nstart = 15)
str(kmeans)
```

```
## List of 9
##  $ cluster     : Named int [1:49] 1 1 1 1 2 1 1 1 2 1 ...
##   ..- attr(*, "names")= chr [1:49] "1" "2" "3" "4" ...
##  $ centers     : num [1:2, 1] -0.232 2.607
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : NULL
##  $ totss       : num 48
##  $ withinss    : num [1:2] 7.28 11.12
##  $ tot.withinss: num 18.4
##  $ betweenss   : num 29.6
##  $ size        : int [1:2] 45 4
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
# Save cluster in the dataframe
x_sub$Cluster <- as.factor(kmeans$cluster)

# Assess a little more descriptively
t1 <- as.table(kmeans$cluster)
t1 <- data.frame(t1)
View(t1)

rownames(t1) <- state$state
colnames(t1)[colnames(t1)=="Freq"] <- "Assignment"
t1$Var1 <- NULL

head(t1, 10) # inspect first 10 obs
```

```
##             Assignment
## Alabama              1
## Alaska               1
## Arizona              1
## Arkansas             1
## California           2
## Colorado             1
## Connecticut          1
## Delaware             1
## Florida              2
## Georgia              1
```

Based on the k-means model, cluster one includes 45 states, and cluster two includes the rest of 4 states.

Fit Gaussian mixture model

```
library(mixtools)
```

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

```
## 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-0518772.
```

```
library(plotGMM)
```

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

```
set.seed(123)
gmm1 <- mvnormalmixEM(xsub_scaled,
            k = 2) # fit the GMM using EM Multivariate and 2 comps
```

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

```
str(gmm1)
```

```
## List of 9
##  $ x         : num [1:49, 1:4] -0.372 -0.229 1.645 -0.804 2.881 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:49] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:4] "t_slength" "slength" "salary_real" "expend"
##   ..- attr(*, "scaled:center")= Named num [1:4] 148 139 55 744
##   .. ..- attr(*, "names")= chr [1:4] "t_slength" "slength" "salary_real" "expend"
##   ..- attr(*, "scaled:scale")= Named num [1:4] 84.1 74 49.4 872.2
##   .. ..- attr(*, "names")= chr [1:4] "t_slength" "slength" "salary_real" "expend"
##  $ lambda    : num [1:2] 0.898 0.102
##  $ mu        :List of 2
##   ..$ : num [1:4] -0.276 -0.245 -0.194 -0.192
##   ..$ : num [1:4] 2.43 2.16 1.71 1.69
##  $ sigma     :List of 2
##   ..$ : num [1:4, 1:4] 0.2453 0.2795 0.2097 0.0311 0.2795 ...
##   ..$ : num [1:4, 1:4] 0.856 1.02 0.398 0.351 1.02 ...
##  $ loglik    : num -75.3
##  $ posterior : num [1:49, 1:2] 1.00 1.00 3.11e-52 1.00 1.82e-100 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:2] "comp.1" "comp.2"
##  $ all.loglik: num [1:26] -417 -159 -152 -136 -129 ...
##  $ restarts  : num 0
##  $ ft        : chr "mvnormalmixEM"
##  - attr(*, "class")= chr "mixEM"
```

```
which(as.data.frame(gmm1$posterior)$comp.1 > 0.5)
```

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

```
which(as.data.frame(gmm1$posterior)$comp.2 > 0.5)
```

```
## [1]  3  5 21 32 38
```

Based on GMM model, cluster one includes 44 states, and cluster two includes the rest of 5 states.

Fit DBSCAN

```
library(dbscan)
```

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

```
dbscan(xsub_scaled, eps = 0.5, minPts = 3)
```

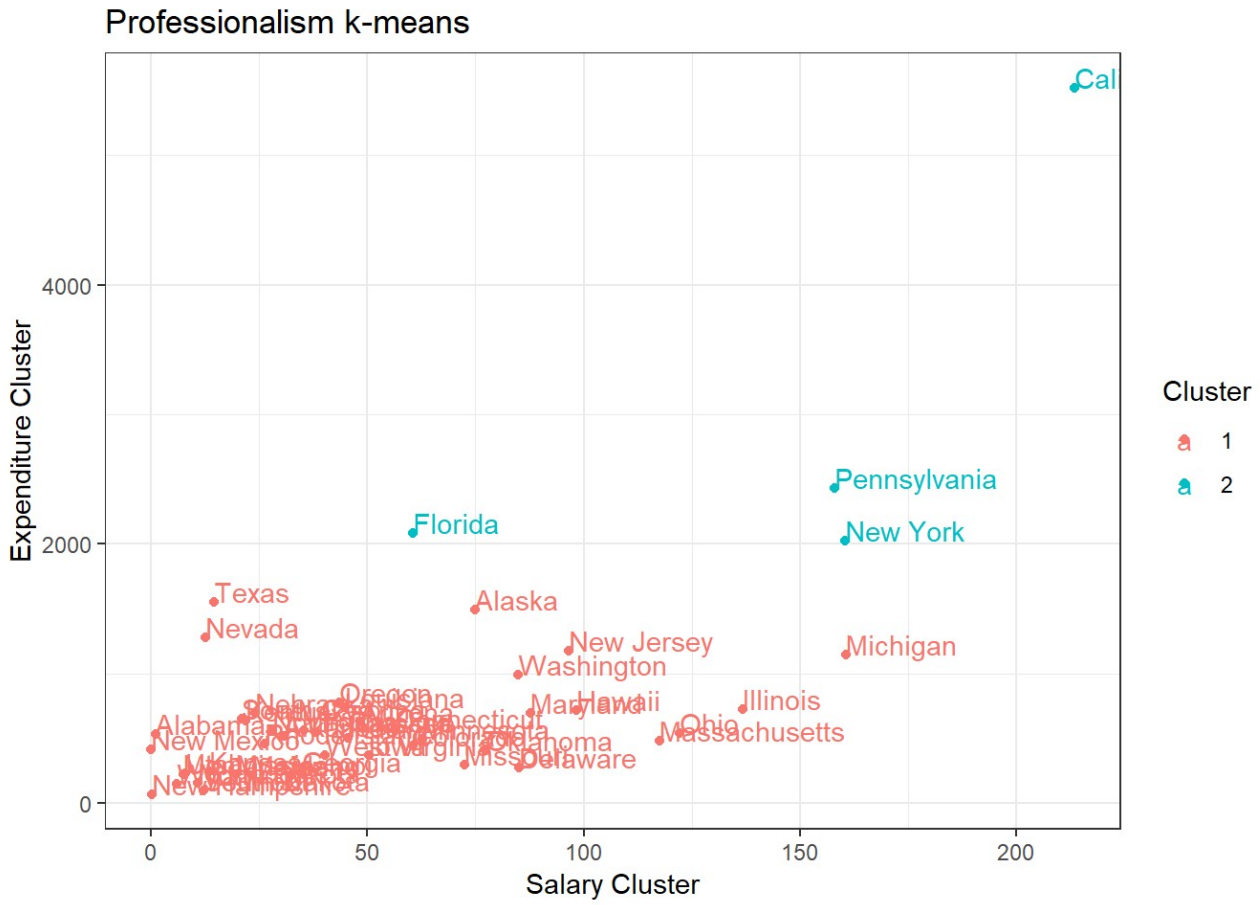
```
## DBSCAN clustering for 49 objects.
## Parameters: eps = 0.5, minPts = 3
## The clustering contains 2 cluster(s) and 14 noise points.
##
##  0  1  2
## 14 32  3
##
## Available fields: cluster, eps, minPts
```

Based on DBSCAN model, 14 of all observations are regarded as noise, while the model returns two clusters, one containing 32 states, the other contains 3.

Clustering outputs

K-means

```
par(mfrow = c(1, 1))

ggplot(x_sub, aes(x = salary_real, y = expend, color = Cluster, label = state$state))
+
  geom_point() +
  geom_text(aes(label = state$state),hjust=0, vjust=0) +
  labs(x = "Salary Cluster",
       y = "Expenditure Cluster",
       title = "Professionalism k-means",
       fill = "Clusters") +
  theme_bw()
```



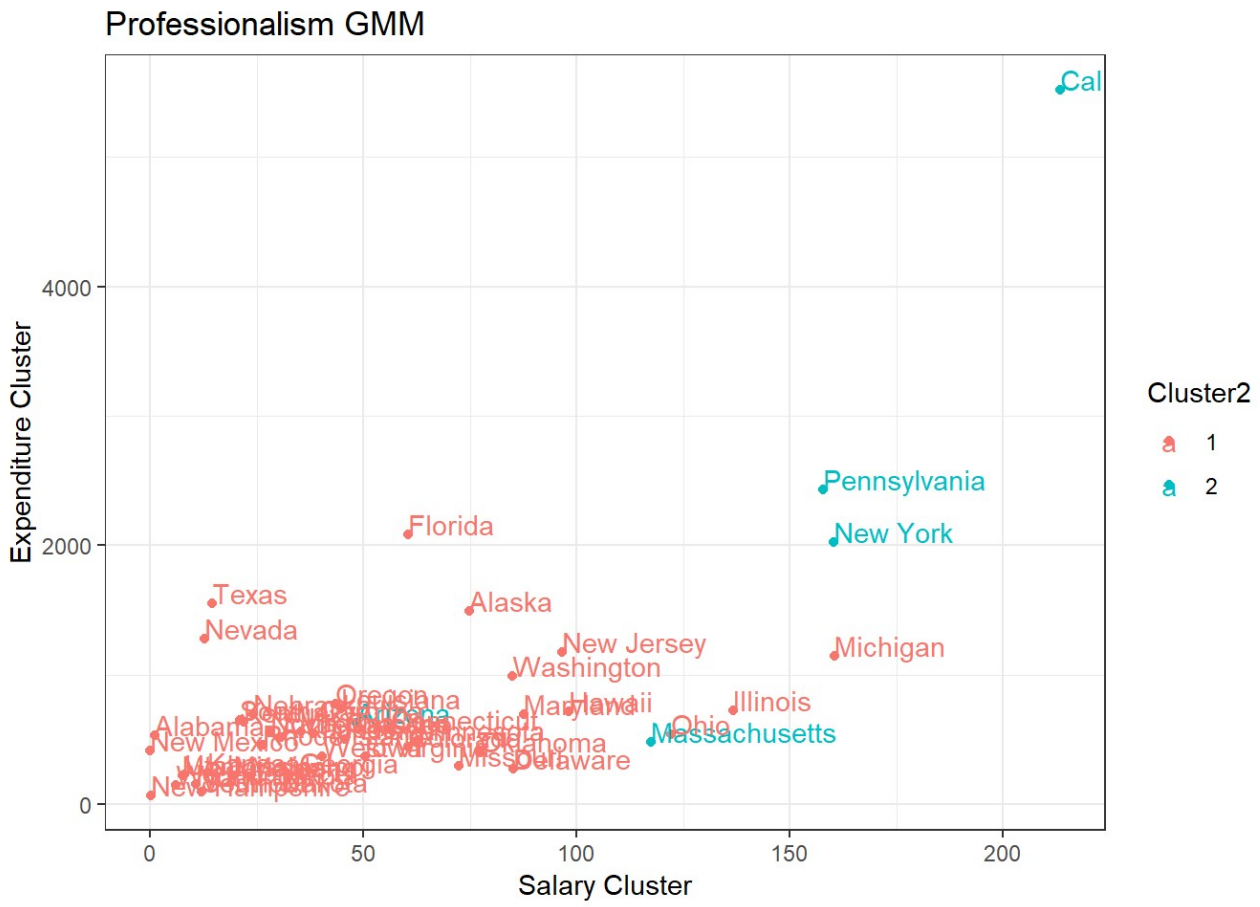
As the scatterplot shows, the four states identified to be cluster two are California, Pennsylvania, Florida and New York.

GMM

```
par(mfrow = c(1, 1))

# create new variable within gmm1

gmm1$cluster <- as.factor(ifelse(as.data.frame(gmm1$posterior)$comp.1 > 0.5, 1, 2))
summary(gmm1$cluster)
```

```
##  1  2
## 44  5
```

```
# Save cluster in the dataframe
x_sub$Cluster2 <- as.factor(gmm1$cluster)

ggplot(x_sub, aes(x = salary_real, y = expend, color = Cluster2)) +
  geom_point() +
  geom_text(aes(label = state$state),hjust=0, vjust=0) +
  labs(x = "Salary Cluster",
       y = "Expenditure Cluster",
       title = "Professionalism GMM",
       fill = "Clusters") +
  theme_bw()
```

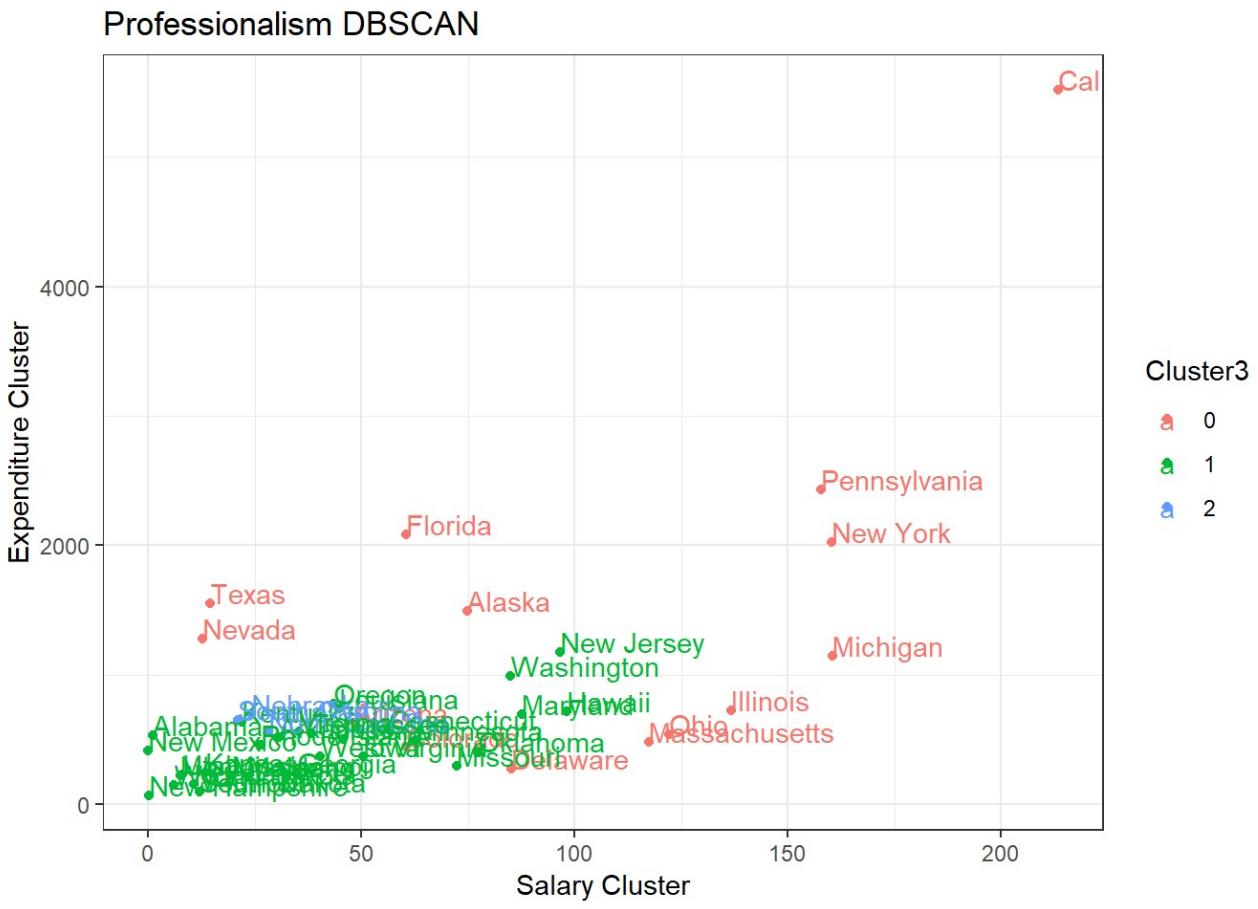


DBSCAN

```
par(mfrow = c(1, 1))

# create new variable within dbscan
dbscan1 <- dbscan(xsub_scaled, eps = 0.5, minPts = 3)
str(dbscan1)
```

```
## List of 3
##  $ cluster: int [1:49] 1 0 0 1 0 0 1 0 0 1 ...
##  $ eps    : num 0.5
##  $ minPts : num 3
##  - attr(*, "class")= chr [1:2] "dbscan_fast" "dbscan"
```

```
# Save cluster in the dataframe
x_sub$Cluster3 <- as.factor(dbscan1$cluster)


ggplot(x_sub, aes(x = salary_real, y = expend, color = Cluster3)) +
  geom_point() +
  geom_text(aes(label = state$state),hjust=0, vjust=0) +
  labs(x = "Salary Cluster",
       y = "Expenditure Cluster",
       title = "Professionalism DBSCAN",
       fill = "Clusters") +
  theme_bw()
```



Internal Validation

```
library(mclust)
```

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

```
## 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:psych':
##
##     sim
```

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

```
x_int <- as.matrix(xsub_scaled[,4])

kmeans_val <- clValid(x_int, 2:10,
                    clMethods = c("kmeans"),
                    validation = "internal"); summary(kmeans_val)
```

```
##
## 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   2.9290  9.4560 11.8444 14.3444 19.0790 23.0147 23.9329 26.442
5 32.4722
##        Dunn           1.3064  0.1175  0.3086  0.3086  0.0599  0.0909  0.1859  0.239
7  0.2616
##        Silhouette     0.8630  0.7124  0.6721  0.6582  0.6048  0.6145  0.6177  0.606
4  0.5890
##
## Optimal Scores:
##
##              Score  Method Clusters
## Connectivity 2.9290 kmeans 2
## Dunn         1.3064 kmeans 2
## Silhouette   0.8630 kmeans 2
```

```
gmm_val <- clValid(x_int, 2:10,
                   clMethods = c("model"),
                   validation = "internal"); summary(gmm_val)
```

```
## Warning in vClusters(mat, clMethods[i], nClust, validation = validation, :
## model unable to find 7 clusters, returning NA for these validation measures
```

```
## Warning in vClusters(mat, clMethods[i], nClust, validation = validation, :
## model unable to find 8 clusters, returning NA for these validation measures
```

```
## Warning in vClusters(mat, clMethods[i], nClust, validation = validation, :
## model unable to find 10 clusters, returning NA for these validation
## measures
```

```
##
## 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   4.6857  6.1615 12.3964 17.3056 33.9603      NA      NA 38.8052
NA
##       Dunn           0.0479  0.0165  0.0095  0.0041  0.0013      NA      NA  0.0053
NA
##       Silhouette     0.6897  0.5109  0.4378  0.4223  0.1726      NA      NA  0.4813
NA
##
## Optimal Scores:
##
##              Score  Method Clusters
## Connectivity 4.6857 model  2
## Dunn         0.0479 model  2
## Silhouette   0.6897 model  2
```

```
internal_val <- clValid(x_int, 2:10,
                    clMethods = c("kmeans", "model"),
                    validation = "internal"); summary(internal_val)
```

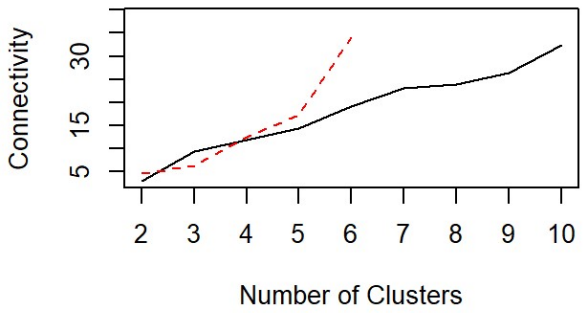
```
## Warning in vClusters(mat, clMethods[i], nClust, validation = validation, :
## model unable to find 7 clusters, returning NA for these validation measures
```

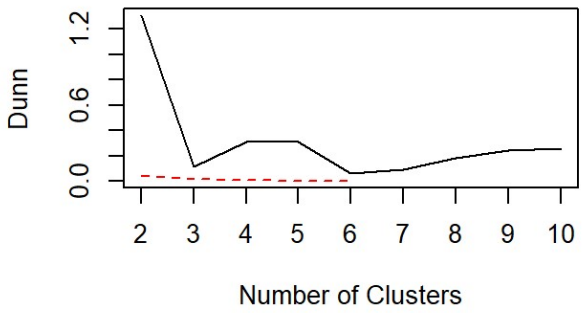
```
## Warning in vClusters(mat, clMethods[i], nClust, validation = validation, :
## model unable to find 8 clusters, returning NA for these validation measures
```

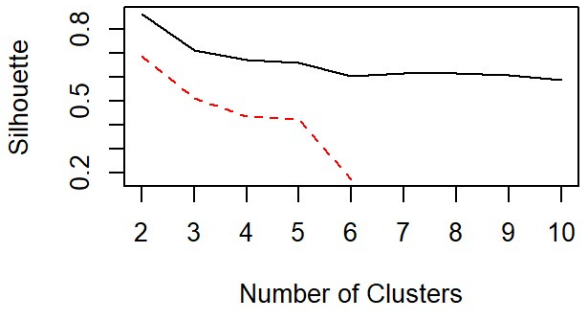
```
## Warning in vClusters(mat, clMethods[i], nClust, validation = validation, :
## model unable to find 10 clusters, returning NA for these validation
## measures
```

```
##
## Clustering Methods:
##  kmeans model
##
## Cluster sizes:
##  2 3 4 5 6 7 8 9 10
##
## Validation Measures:
##                            2       3       4       5       6       7       8
9      10
##
## kmeans Connectivity   2.9290  9.4560 11.8444 14.3444 19.0790 23.0147 23.9329 26.442
5 32.4722
##        Dunn           1.3064  0.1175  0.3086  0.3086  0.0599  0.0909  0.1859  0.239
7  0.2616
##        Silhouette     0.8630  0.7124  0.6721  0.6582  0.6048  0.6145  0.6177  0.606
4  0.5890
## model  Connectivity   4.6857  6.1615 12.3964 17.3056 33.9603      NA      NA 38.805
2      NA
##        Dunn           0.0479  0.0165  0.0095  0.0041  0.0013      NA      NA  0.005
3      NA
##        Silhouette     0.6897  0.5109  0.4378  0.4223  0.1726      NA      NA  0.481
3      NA
##
## Optimal Scores:
##
##              Score  Method Clusters
## Connectivity 2.9290 kmeans 2
## Dunn         1.3064 kmeans 2
## Silhouette   0.8630 kmeans 2
```

```
par(mfrow = c(2, 2))
plot(internal_val, legend = FALSE,
     type = "l",
     main = " ")
```







K_means model returns a Dunn index of 1.3 and high silhouette width of 0.8630 for 2 clusters; whereas GMM returns a Dunn index of 0.0479, and a silhouette width is 0.6897 for 2 clusters.

According to the silhouette width produced by different algorithm methods, k-means is able to provide a better configuration of observations within the cluster; regardless of the clustering algorithm, the optimal number of clusters seems to be two using the three measures.

Even though we have the support of validation statistics, we might still run into the risk of misrepresenting the data depending on the partitioning method we choose because 1) while a good statistic suggests one particular method to yield best result, such clustering may not be as consistent and stable as we would hope to see; 2) there could be other dimensions that are potentially useful to be added into the feature space; 3) we might fit a gaussian model into a non-normal distribution of parameters.