Cao - Problem Set 3

Steven Cao

10/25/2019

Problem 1

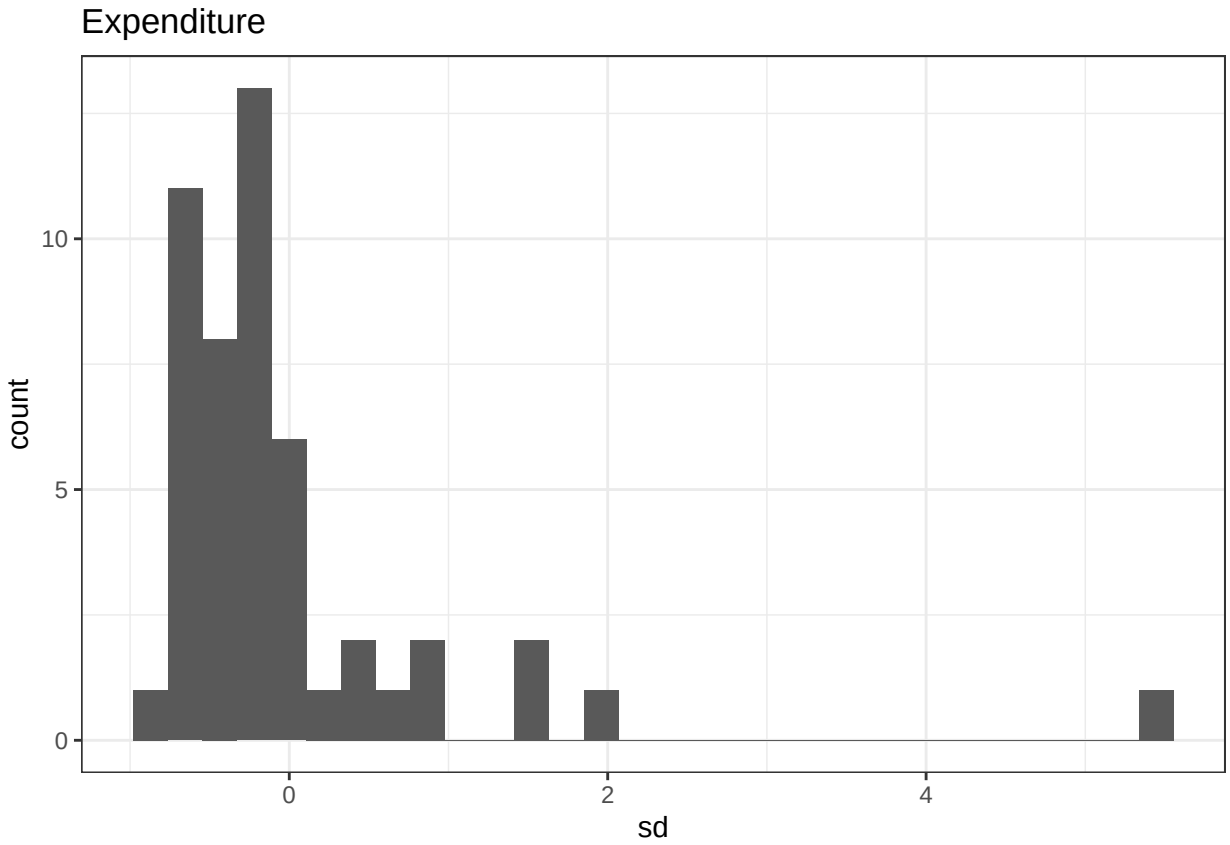
The general background information is that proxy measures such as amount of spending on state legislators are indicators of how "professional" they are in doing their job, with the ideal sense of professionality being Congress.

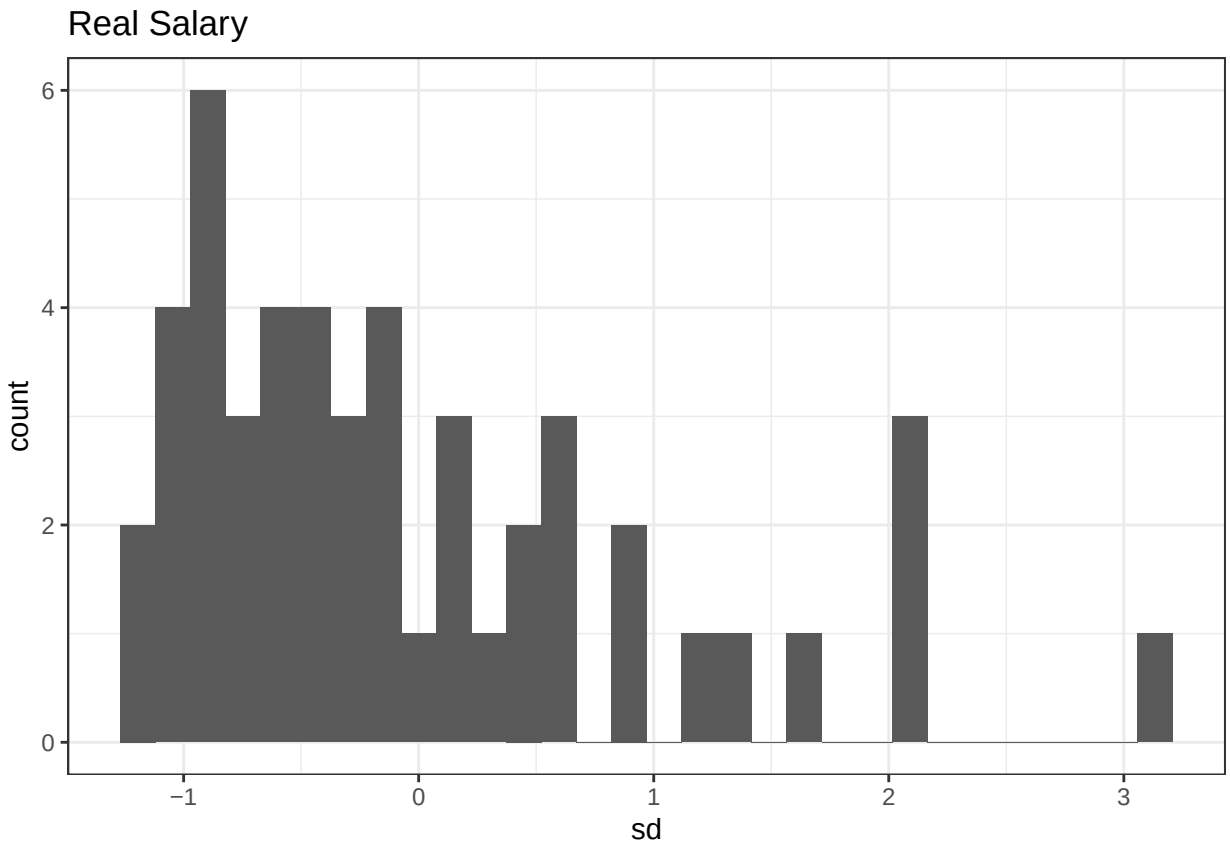
Problem 2

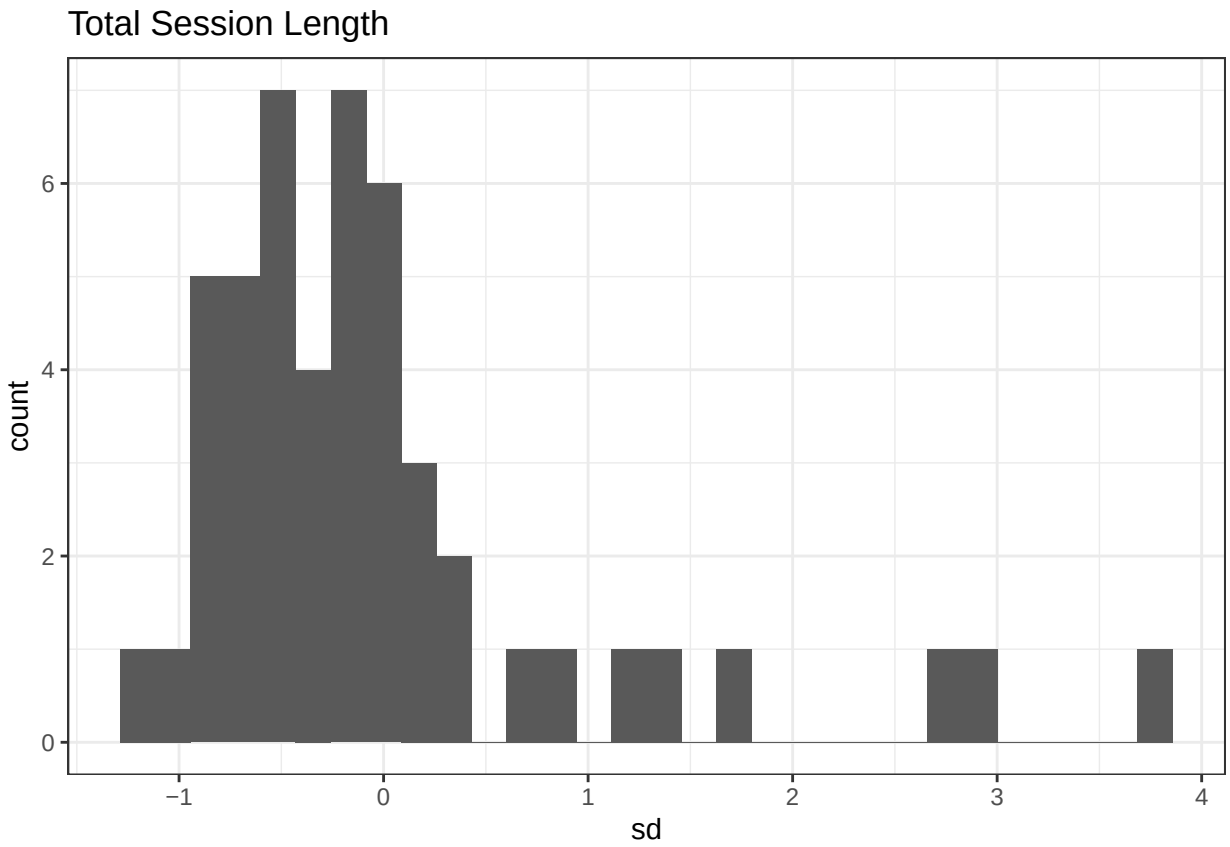
```
professionalityData <- x %>%
  dplyr::filter(sessid=="2009/10") %>% # only for the most recent observation of all states
  dplyr::transmute( state=state, # select only the features of interest and standardise them as appropr
                    expend=scale(expend),
                    salary_real=scale(salary_real),
                    t_slength=scale(t_slength)) %>%
  na.omit() # drop-kick states with missing data, they weren't professional anyway (in this case, Wisco

professionalityData_States <- professionalityData %>%
  dplyr::select(state)

professionalityData <- professionalityData %>%
  dplyr::select(expend, salary_real, t_slength) %>%
  as.data.frame.matrix()
```

Problem 3

```
## Skim summary statistics
##  n obs: 49
##  n variables: 3
##
## -- Variable type:numeric ------------------------------------------------------------------------
##     variable missing complete  n     mean   sd    p0   p25   p50    p75
##       expend       0       49 49 -0.002   1.01 -0.78 -0.54 -0.24 -0.025
##  salary_real       0       49 49 -0.019   1    -1.13 -0.73 -0.32  0.44
##    t_slength       0       49 49 -7.8e-17 1    -1.28 -0.6  -0.24  0.13
##  p100     hist
##  5.53 <U+2587><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581>
##  3.19 <U+2587><U+2585><U+2583><U+2582><U+2581><U+2581><U+2581><U+2581>
##  3.69 <U+2583><U+2587><U+2583><U+2581><U+2581><U+2581><U+2581><U+2581>
```

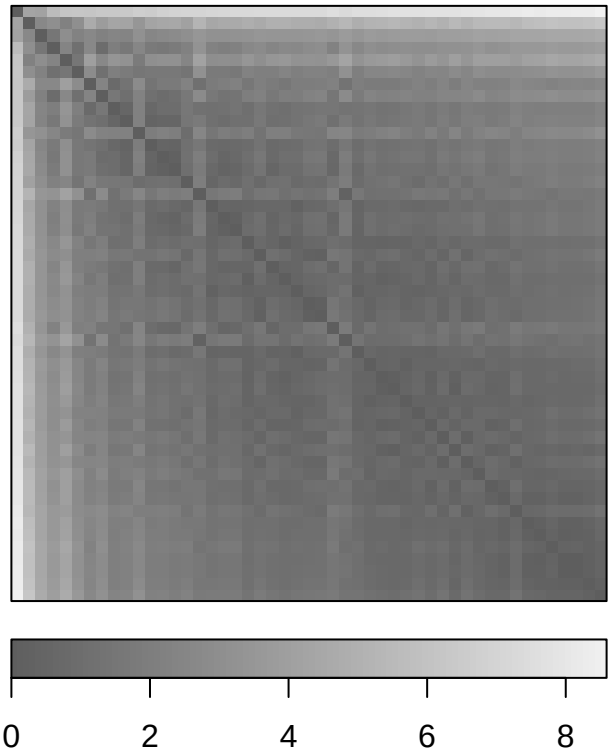

Expenditure
count
10
5
0
0
2
4
sd

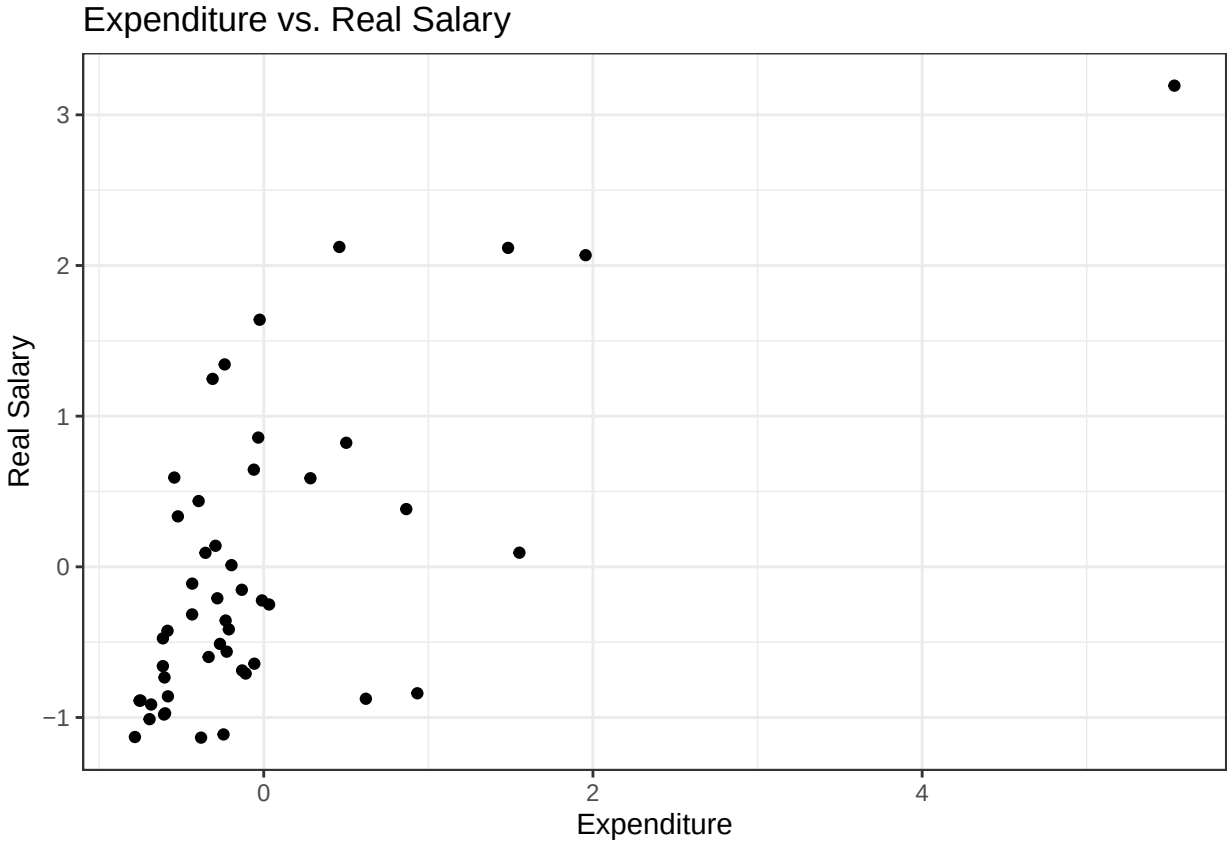

Real Salary
count
sd
0
2
4
6
−1
0
1
2
3

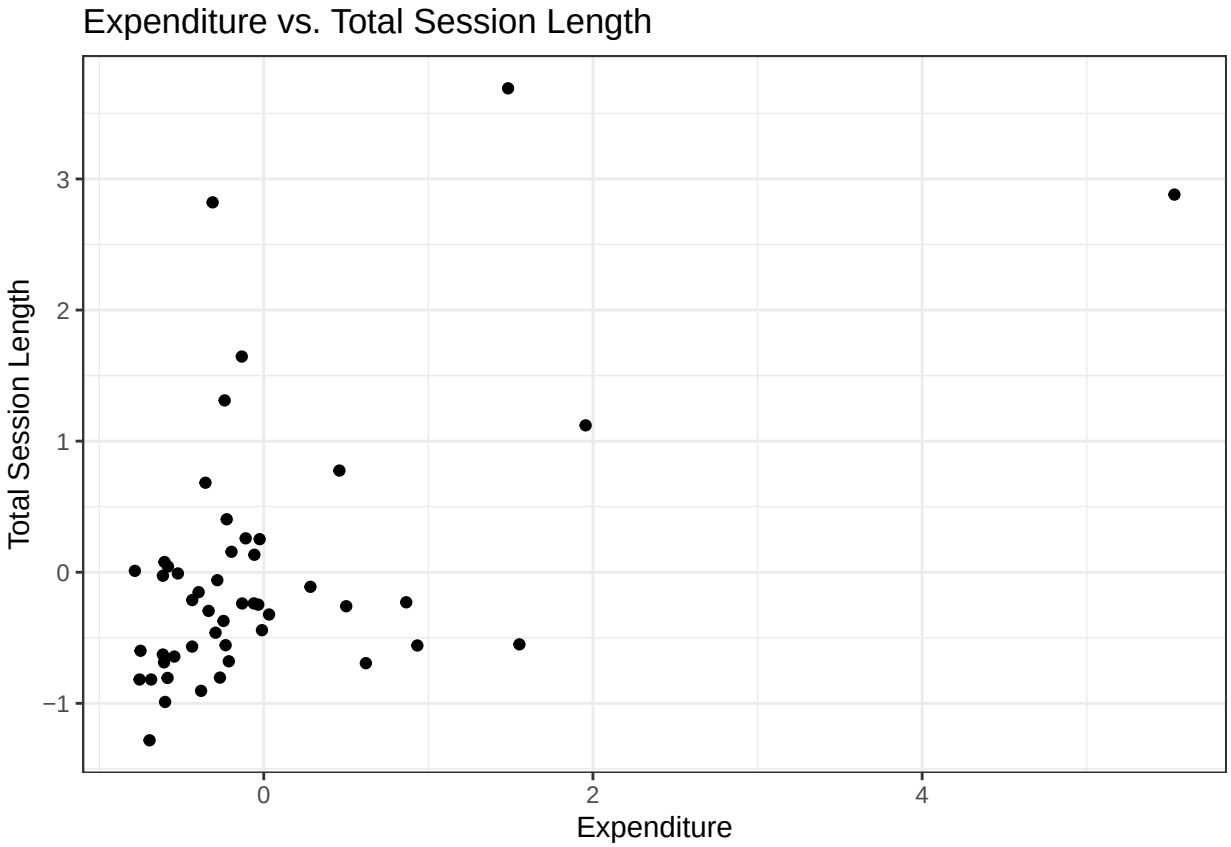


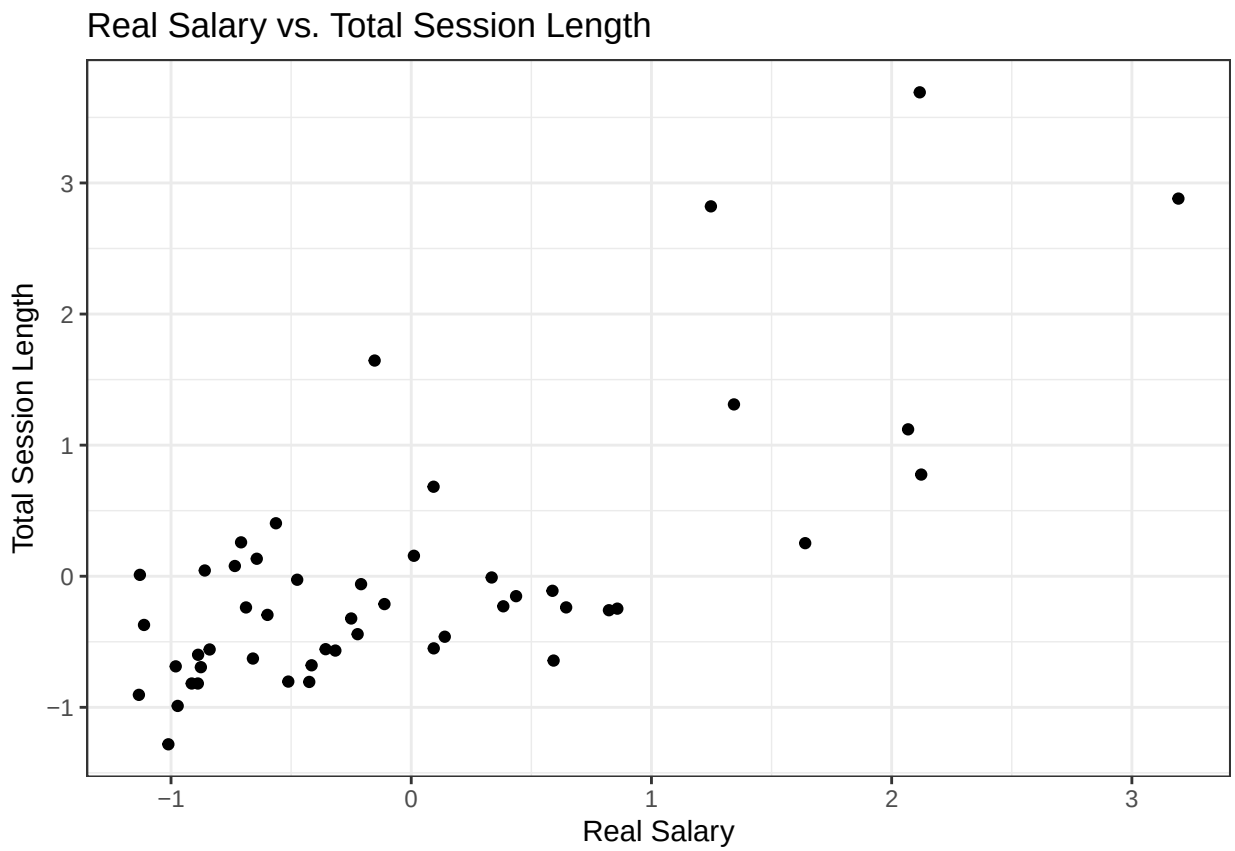
All three variables - expenditure, salary, and total session length - seem to be unimodal with relatively long rightward skews. The expenditure variable appears to have an outlier.

Problem 4


0
2
4
6
8


Expenditure vs. Real Salary
Real Salary
3
2
1
0
−1
0
2
4
Expenditure

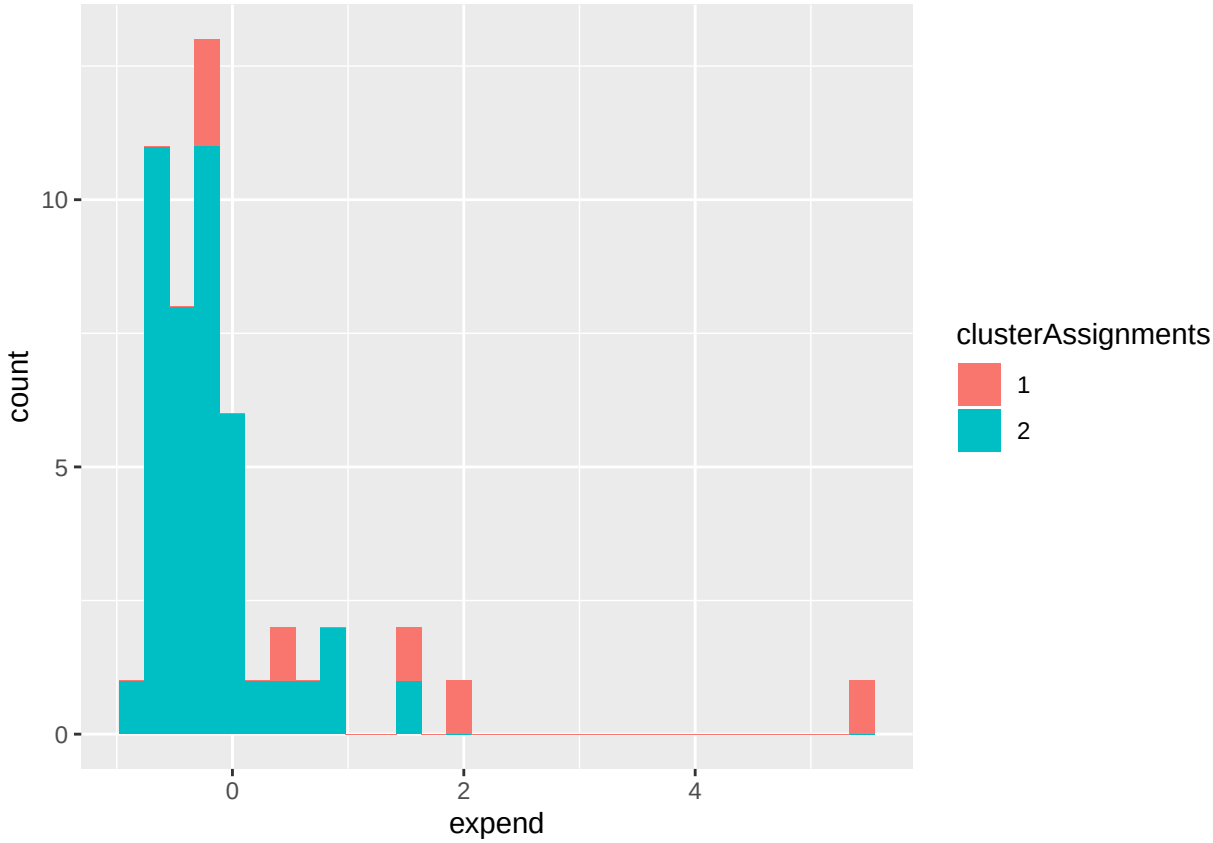

Expenditure vs. Total Session Length
Total Session Length
3
2
1
0
−1
0
2
4
Expenditure



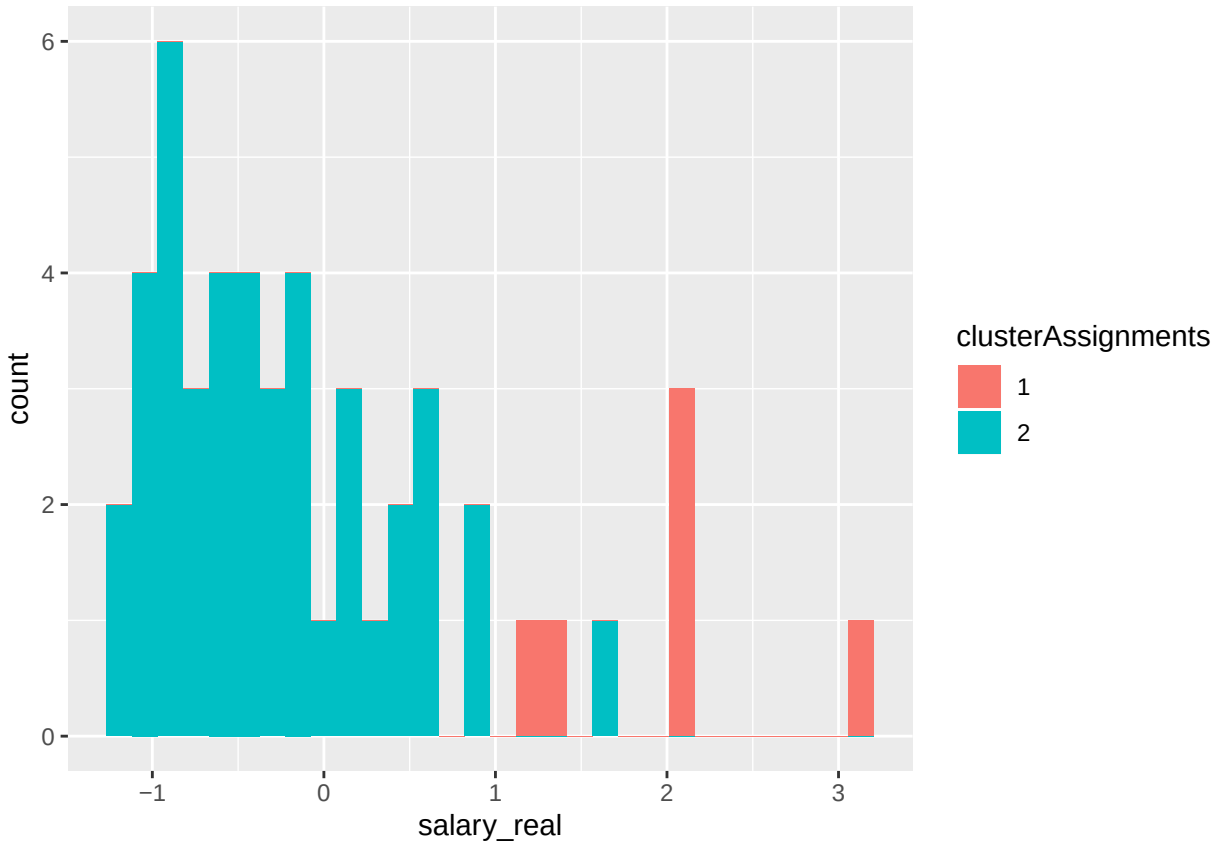
The potential for clustering here seems very poor. At best, the ODI and scatterplots suggest that data *gradually* gets less dense the further we go from the point of origin, but nothing suggests that the data will group nicely (if at all) into clusters.

Problem 5

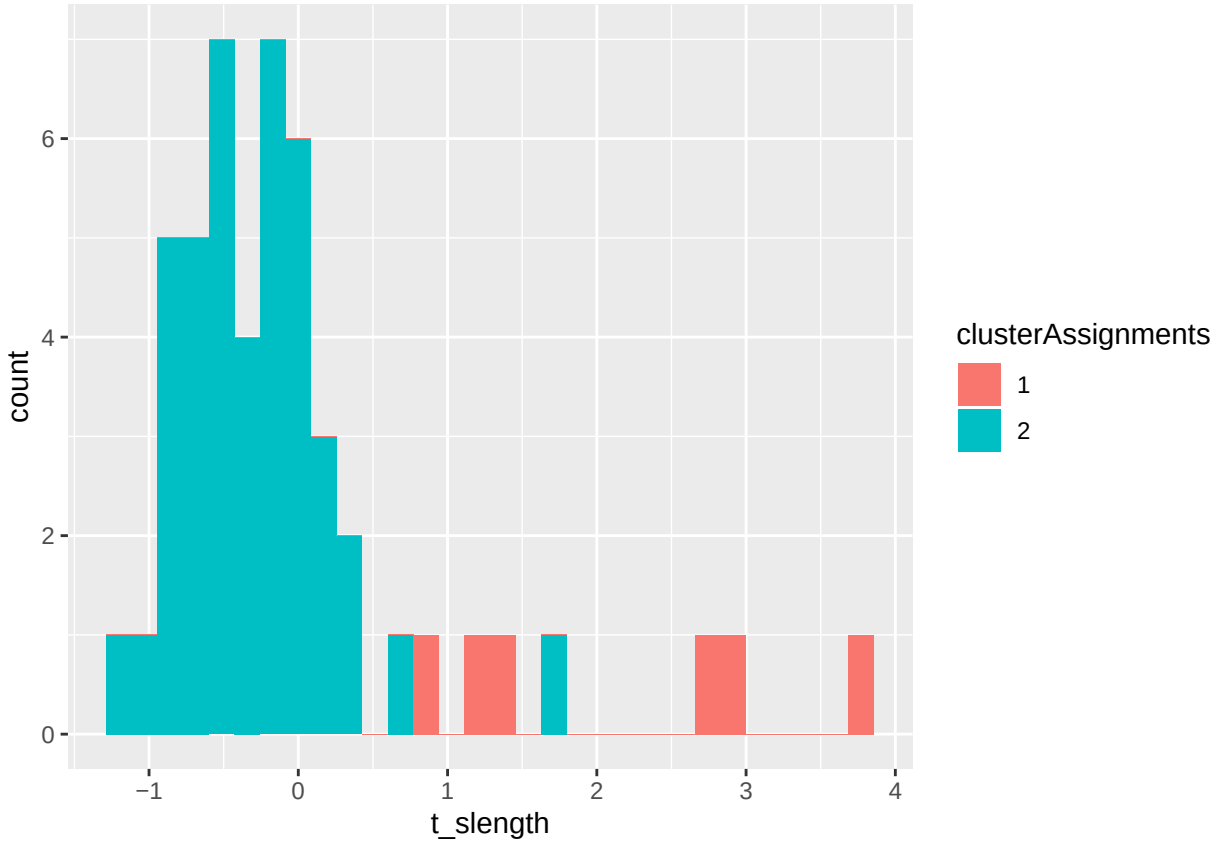
```
kmeans_professionalityData <- kmeans(professionalityData, centers=2, nstart=10)
clusterAssignments_professionalityData <- as.data.frame(kmeans_professionalityData$cluster)
professionalityData$clusterAssignments = ifelse(clusterAssignments_professionalityData==1,"1","2")

ggplot(professionalityData, aes(expend, fill = clusterAssignments)) + geom_histogram(bins=30)
```




```
ggplot(professionalityData, aes(salary_real, fill = clusterAssignments)) + geom_histogram(bins=30)
```




```
ggplot(professionalityData, aes(t_slength, fill = clusterAssignments)) + geom_histogram(bins=30)
```



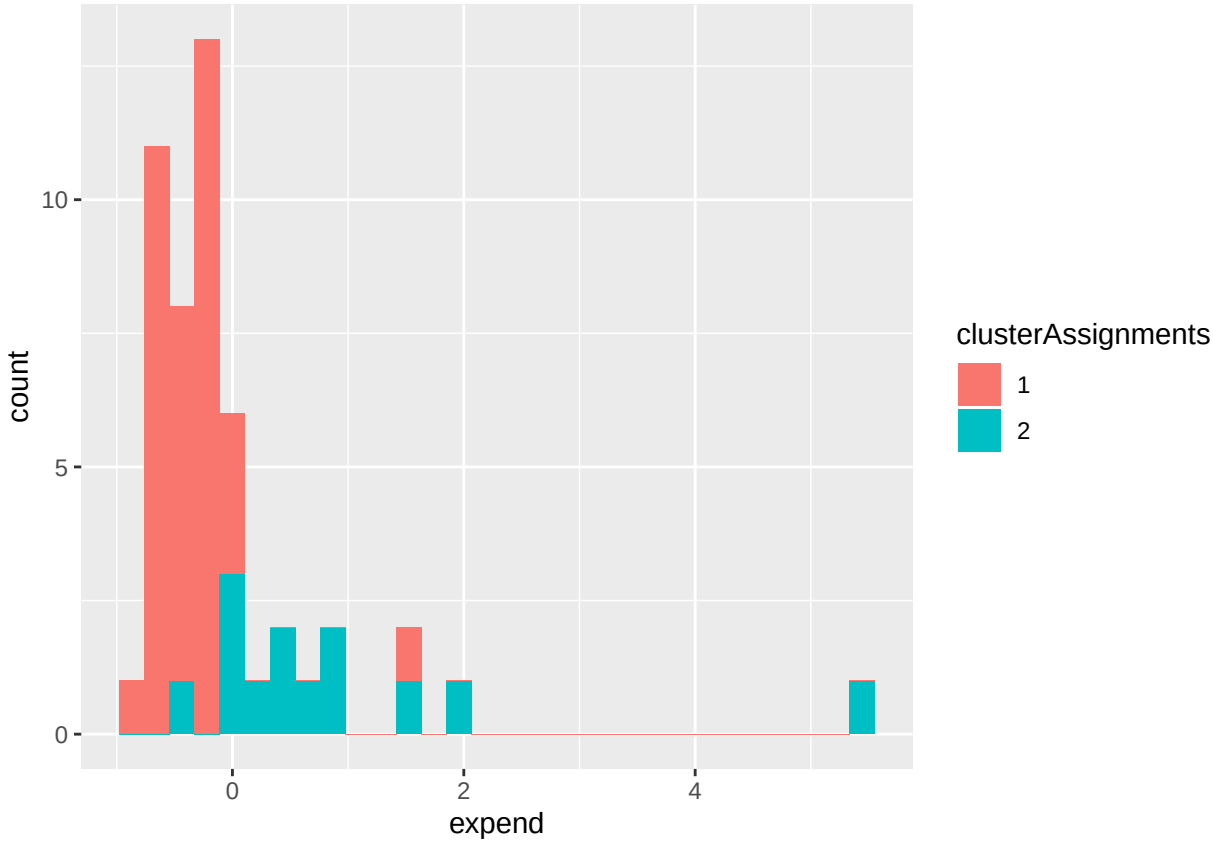
Most of the observations (i.e. the ones surrounding the mean) are grouped into one cluster for all three variables. The rest of the observations (i.e. the ones spread further out and located in the tail) are, for the most part, grouped into the second cluster for all three measures.

Problem 6

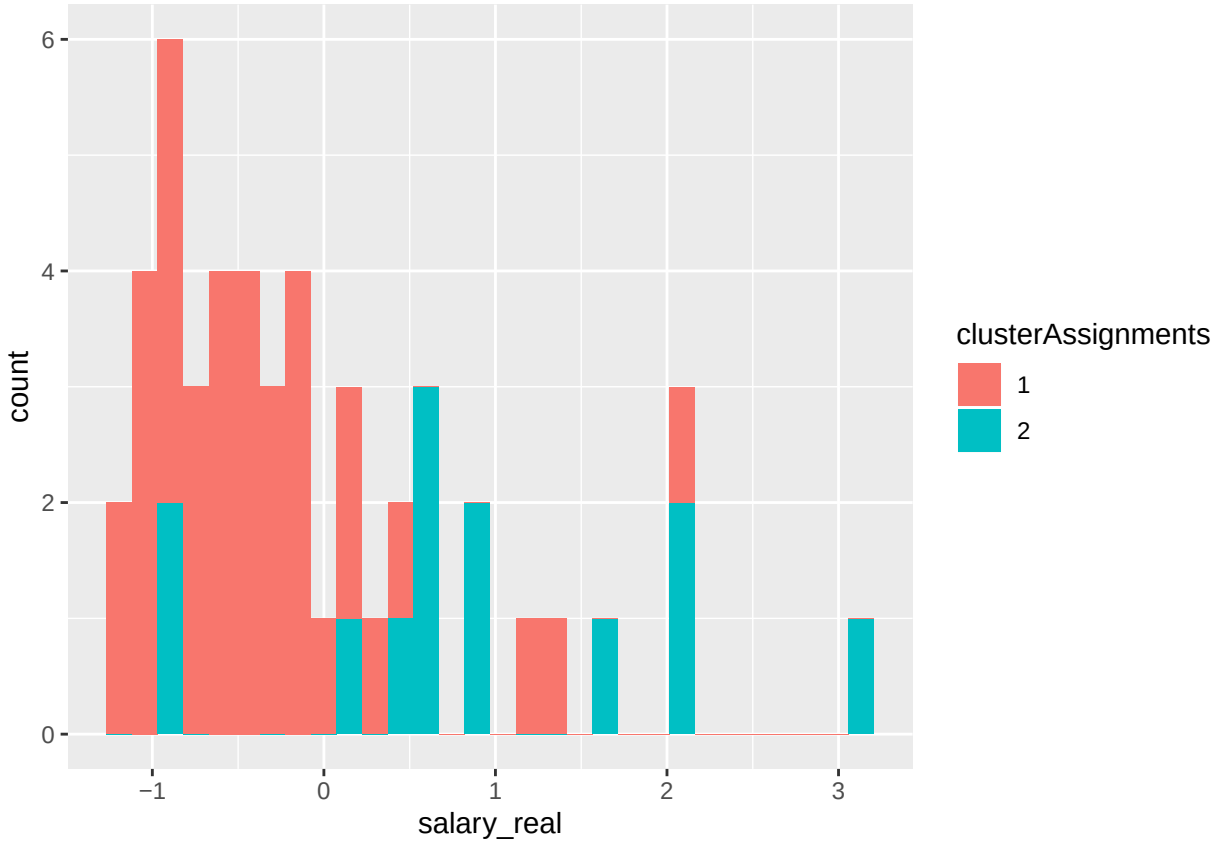
```
gmm_professionalityData <- mvnormalmixEM(professionalityData, k=2)
```

```
## number of iterations= 42
```

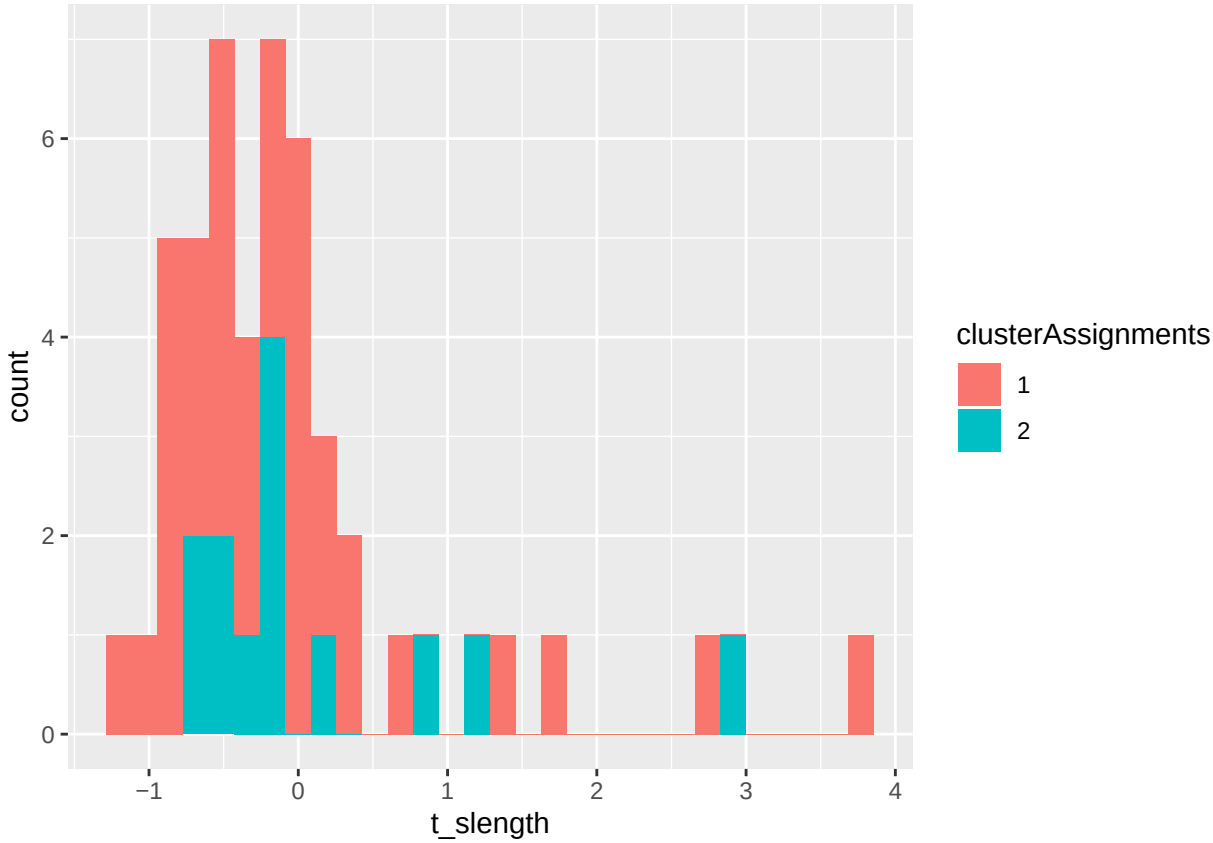
```
gmmAssignments_professionalityData <- gmm_professionalityData[["posterior"]][,1] < gmm_professionalityDa
professionalityData$clusterAssignments = ifelse(gmmAssignments_professionalityData==TRUE,"1","2")

ggplot(professionalityData, aes(expend, fill = clusterAssignments)) + geom_histogram(bins=30)
```




```
ggplot(professionalityData, aes(salary_real, fill = clusterAssignments)) + geom_histogram(bins=30)
```



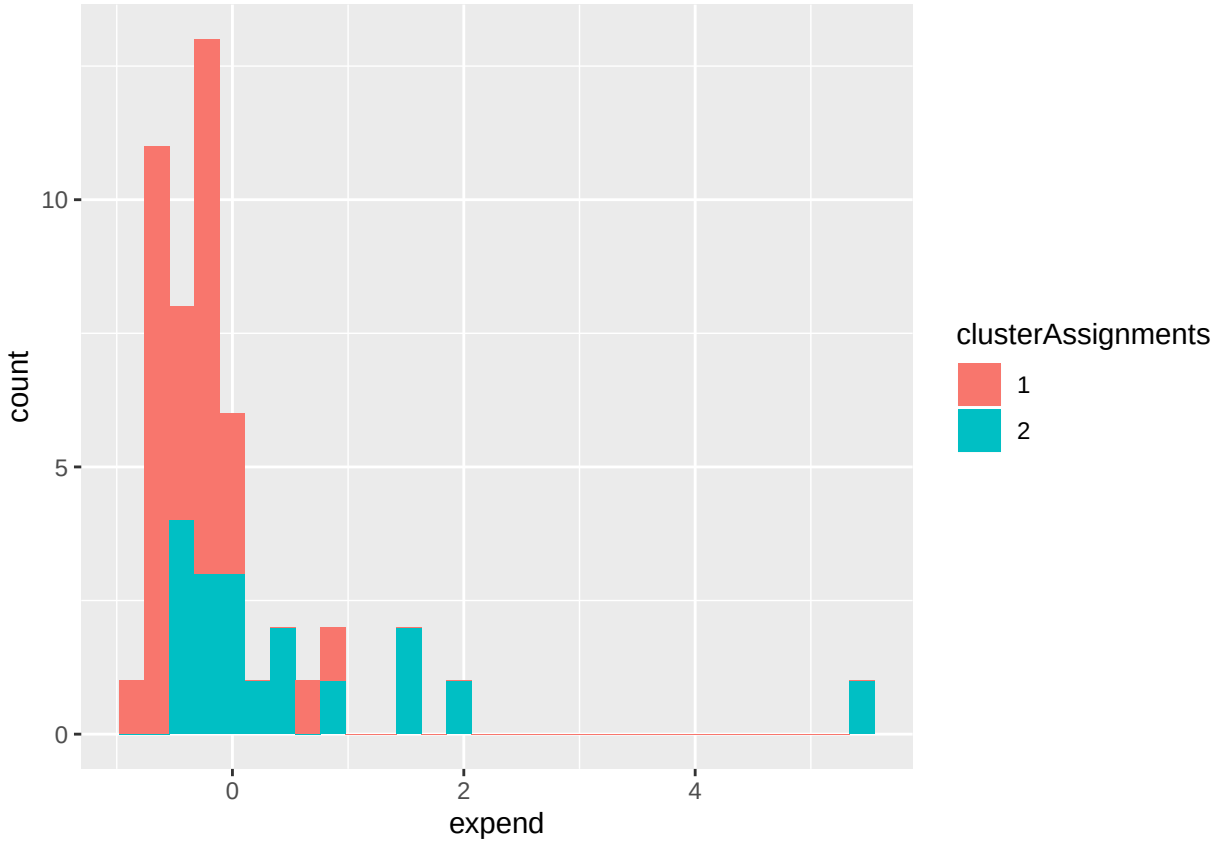

```
ggplot(professionalityData, aes(t_slength, fill = clusterAssignments)) + geom_histogram(bins=30)
```



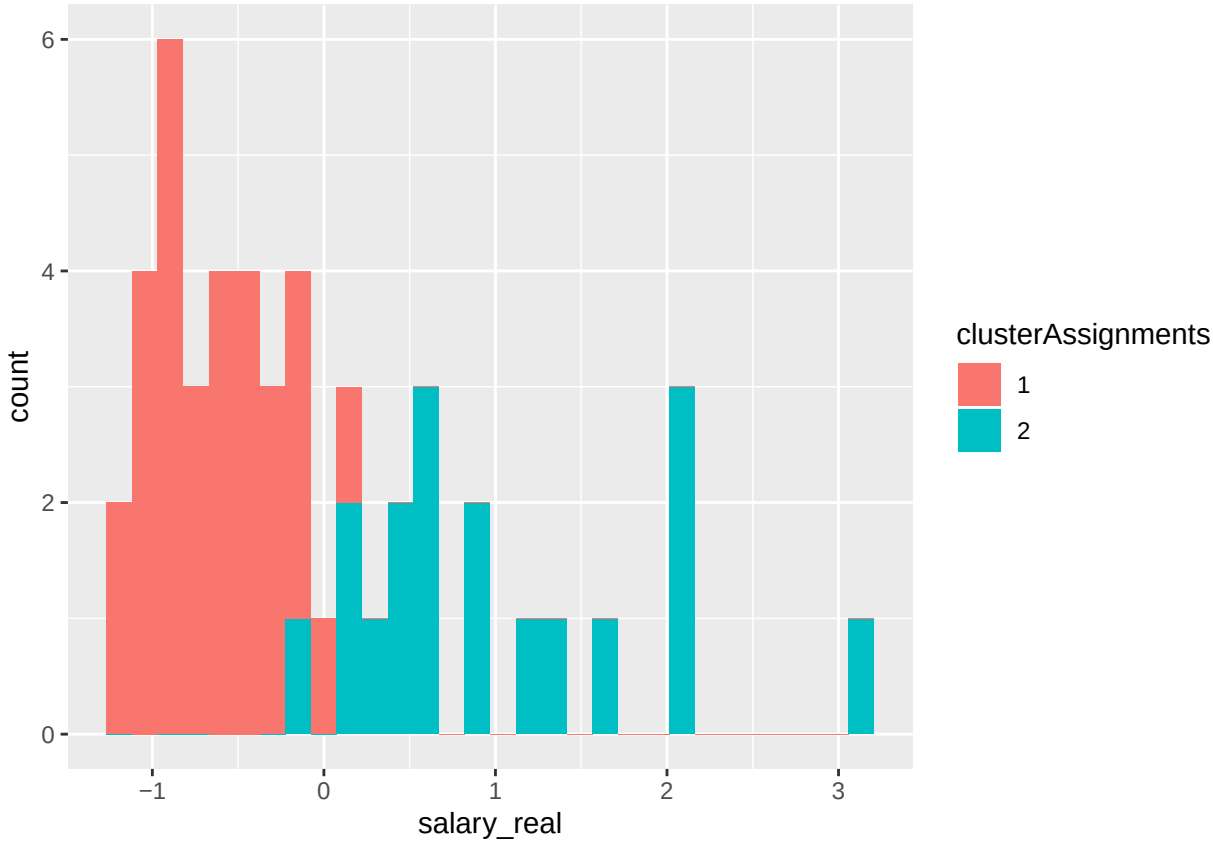
The groupings are much more mixed. While the first grouping is still based mainly around the mean for all three variables, observations that are found in the tail still get clustered into the first group. Likewise, some observations that are found near the mean are grouped into the second cluster for all three variables - in particular, the total session length variable.

Problem 7

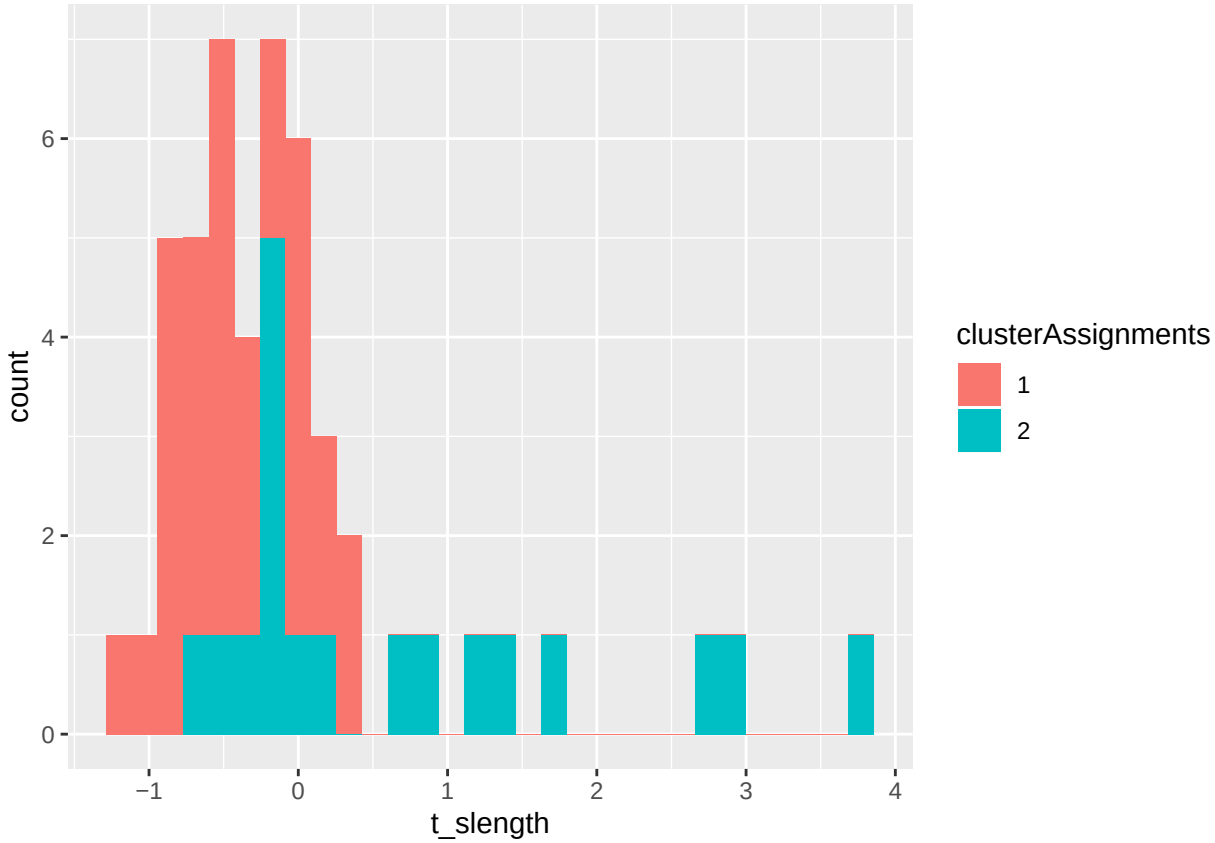
```
kmedoids_professionalityData = pam(professionalityData, k=2)
#view(kmedoids_professionalityData)
clusterAssignments_professionalityData <- as.data.frame(kmedoids_professionalityData$cluster)
professionalityData$clusterAssignments = ifelse(clusterAssignments_professionalityData==1,"1","2")

ggplot(professionalityData, aes(expend, fill = clusterAssignments)) + geom_histogram(bins=30)
```




```
ggplot(professionalityData, aes(salary_real, fill = clusterAssignments)) + geom_histogram(bins=30)
```

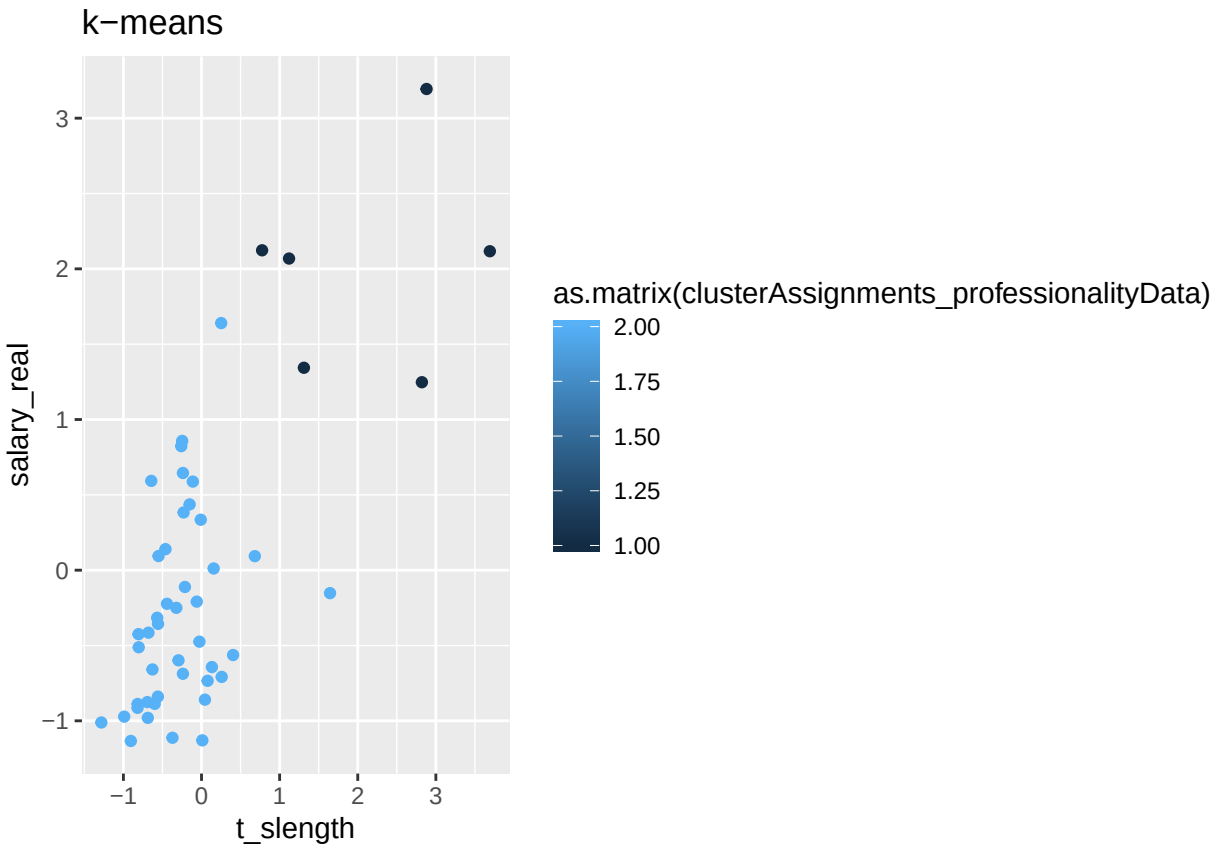


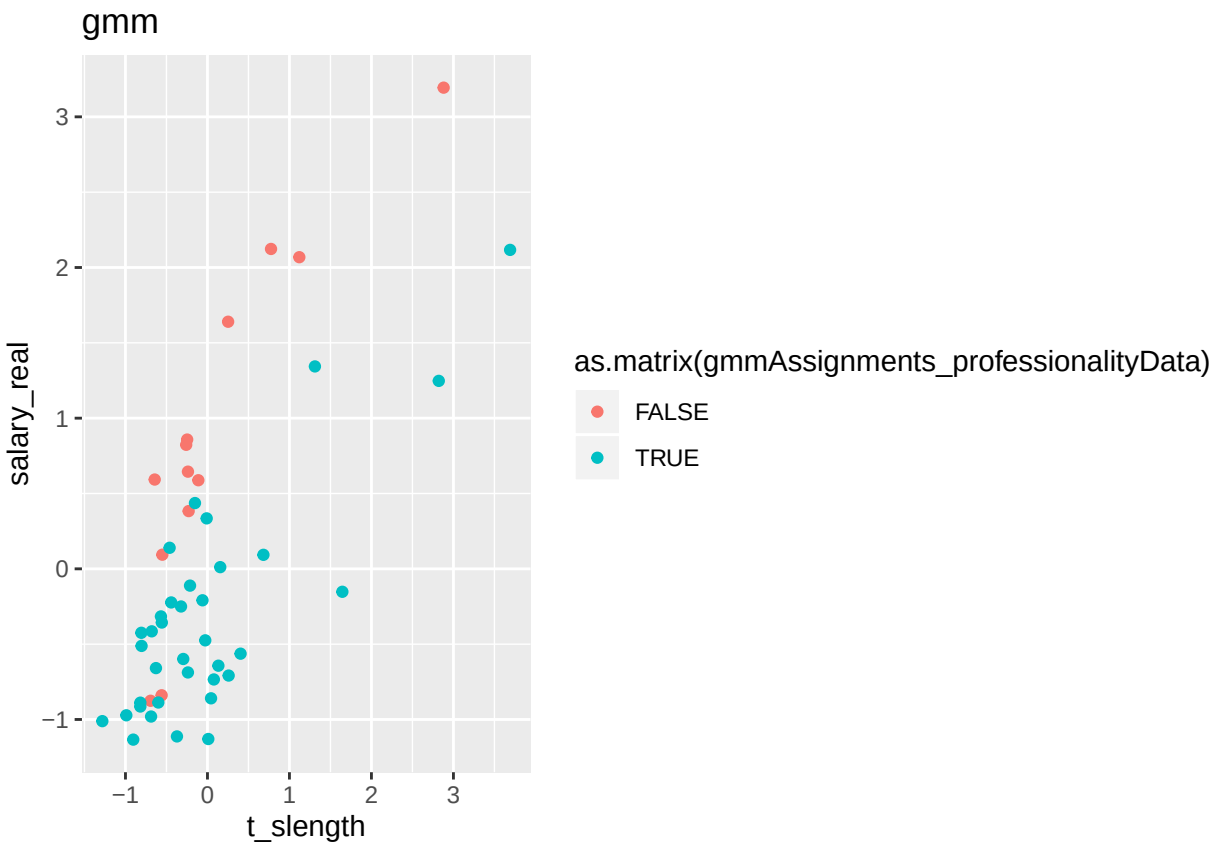

```
ggplot(professionalityData, aes(t_slength, fill = clusterAssignments)) + geom_histogram(bins=30)
```

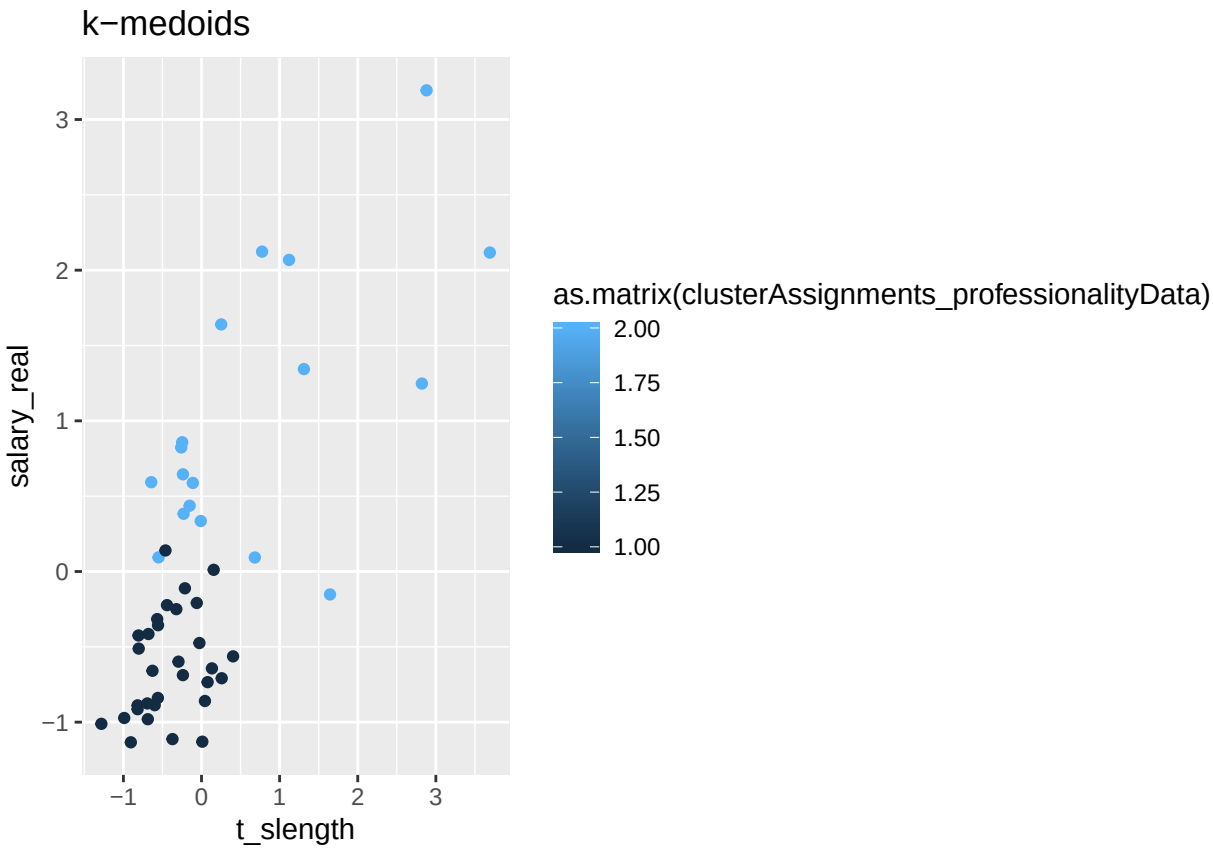


The clustering method used was PAM (aka k-medoids), and this yielded the most interesting clusterings. For **expenditure**, group 1 clustering was found mostly near the mean, while group 2 clustering was also found near the mean, but also included most of the tail (and the one outlier). For **salary**, the locale around the mean was almost exclusively dominated by group 1, while the tail was virtually exclusive to group 2. For **total session length**, group 1 clustering was found only around the mean, while group 2 clustering was found both around the mean, while also comprising the entirety of the tail.

Problem 8


k−means
salary_real
t_slength
as.matrix(clusterAssignments_professionalityData)
2.00
1.75
1.50
1.25
1.00


gmm
salary_real
t_slength
as.matrix(gmmAssignments_professionalityData)
FALSE
TRUE



The **k-means** scatterplot seems to indicate that observations were separated into whether they were clustered around the dense area, or whether they were far away from it all. The **GMM** scatterplot, on the other hand, doesn't seem to be entirely straightforward nor intuitive in what it's trying to cluster. The **k-medoids** scatterplot seems to be grouping like k-means, except it's putting more of the observations on the periphery of the dense cluster into the other grouping instead.

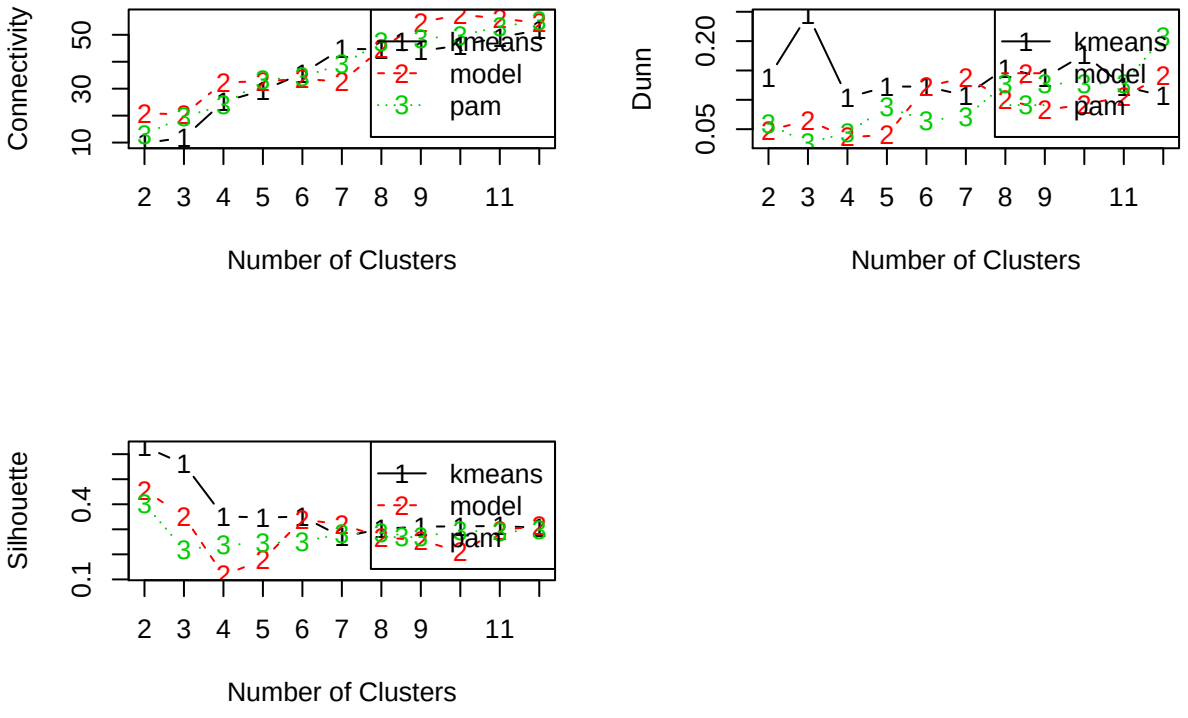
Problem 9

```
validation_professionalityData <- clValid( as.matrix(professionalityData), 2:12, clMethods=c("kmeans","
summary(validation_professionalityData)
```

```
##
## Clustering Methods:
##  kmeans model pam
##
## Cluster sizes:
##  2 3 4 5 6 7 8 9 10 11 12
##
## Validation Measures:
##                             2       3       4       5       6       7       8       9      10      11
##
## kmeans Connectivity   9.7976 11.8151 25.1294 29.3988 35.6790 44.7714 44.6175 44.0183 46.0389 49.0651
##        Dunn           0.1376  0.2451  0.1037  0.1226  0.1226  0.1072  0.1536  0.1380  0.1773  0.1231
##        Silhouette     0.6301  0.5606  0.3516  0.3471  0.3505  0.2696  0.3019  0.3108  0.3117  0.3133
```

```
## model  Connectivity  20.9008 20.4266 32.3556 32.7306 33.6480 32.6036 44.8750 54.6333 57.4885 56.0671
##        Dunn           0.0482  0.0645  0.0365  0.0399  0.1226  0.1383  0.1009  0.0828  0.0918  0.1055
##        Silhouette     0.4560  0.3521  0.1172  0.1774  0.3392  0.3197  0.2680  0.2566  0.2095  0.2965
## pam    Connectivity  12.9567 19.6270 23.8060 33.1992 34.4365 39.0310 47.2155 48.5488 49.7135 53.0048
##        Dunn           0.0588  0.0263  0.0440  0.0874  0.0650  0.0712  0.1277  0.1277  0.1277  0.1277
##        Silhouette     0.4012  0.2194  0.2372  0.2451  0.2504  0.2823  0.2851  0.2709  0.2945  0.2893
##
## Optimal Scores:
##
##              Score  Method Clusters
## Connectivity 9.7976 kmeans 2
## Dunn         0.2451 kmeans 3
## Silhouette   0.6301 kmeans 2
```

```
par(mfrow = c(2,2))
plot(validation_professionalityData, main=" ")
```



Keeping in mind the general rule of thumb that a lower connectivity value is better, whereas a higher silhouette/dunn value is better, the first partitioning method (i.e. **k-means**) seems to be the best out of the methods, which is particularly evident when the number of clusters is 2 (all three methods tend to converge towards 12). The other two methods seem roughly similar to each other in terms of overall "performance".

Problem 10

The "normal" behaviour for most states is to pay their legislators not very much, but also for those same legislators to not spend as much time in a session. Other states, however (I suspect the more populated

ones, i.e. the ones with cities), seem to be paid more but also spend more of their time in a session.

The **k-means** approach is the most optimal of them all, and generally it works best at k=2. (Although, for it's best at k=3 for the Dunn index *only*; here it is worse on the other validation measures.)

A "sub-optimal" partitioning method may make sense depending on the nature of the data. If we can imagine continuous data - in which someone or some observation may fall into more than two groups, or may fall in between them - then GMM may fare better. Even if it still does worse (like it did here), GMM does preserve/provide more information, and therefore more room for interpretation. (An example of where GMM might work well where no hard partitioning method would: distinguishing two sample populations which have the exact same means but extremely different variances.) Hard partitioning like k-means might make sense if we have a hunch that there are categorical differences underlying the data.