Assignment3_AbhishekPandit

Abhishek Pandit

25 October 2019

We will be using mostly base R with a few specialized libraries:

Now the data

```
legprof<- read.csv('leg_prof.csv') #loaded from RWorkspace using wroite.csv
set.seed(1984)
```

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)

First, we explore some of the variables

```
unique(legprof$year)
```

```
##  [1] 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000
## [15] 2002 2004 2006 2008 2010 1975 1977 1979 1981 1983 1985 1987 1989 1991
## [29] 1993 1995 1997 1999 2001 2003 2005 2007 2009 2011
```

```
unique(legprof$sessid)
```

```
##  [1] 1973/4  1975/6  1977/8  1979/80 1981/2  1983/4  1985/6  1987/7
##  [9] 1989/90 1991/2  1993/4  1995/6  1997/8  1999/00 2001/2  2003/4
## [17] 2005/6  2007/8  2009/10
## 19 Levels: 1973/4 1975/6 1977/8 1979/80 1981/2 1983/4 1985/6 ... 2009/10
```

As expected, we see years at intervals of 2. Now, onto the munging. We will need states later for plotting purposes. So it would make sense to save the column separately for now and then reunite it with the main data frame.

```
legprof.all <- legprof %>%
  filter(sessid=="2009/10") %>%
  dplyr::select(expend, t_slength, slength, salary_real, stateabv)%>%
  na.omit() %>%
  mutate_if(is.numeric, scale)%>%
  as.data.frame()
```

```
## Warning: package 'bindrcpp' was built under R version 3.5.2
```

```
legprof.state<- legprof.all[,"stateabv"]

legprof_sub <-legprof.all %>%
  dplyr::select(expend, t_slength, slength, salary_real)%>%
  as.data.frame()

head(legprof.all)
```

```
##       expend  t_slength    slength salary_real stateabv
## 1 -0.2399910 -0.3716599 -0.4594723  -1.0920009       AL
## 2  0.8591198 -0.2294089 -0.1452309   0.4011333       AK
## 3 -0.1299408  1.6453067  0.7951955  -0.1335656       AZ
## 4 -0.2612061 -0.8036462 -0.7881756  -0.4923902       AR
## 5  5.4785453  2.8807257  1.7767099   3.2069914       CA
## 6 -0.3485530  0.6827338  0.9008887   0.1113595       CO
```

So we have one value for almost every state

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

```
summary(legprof_sub)
```

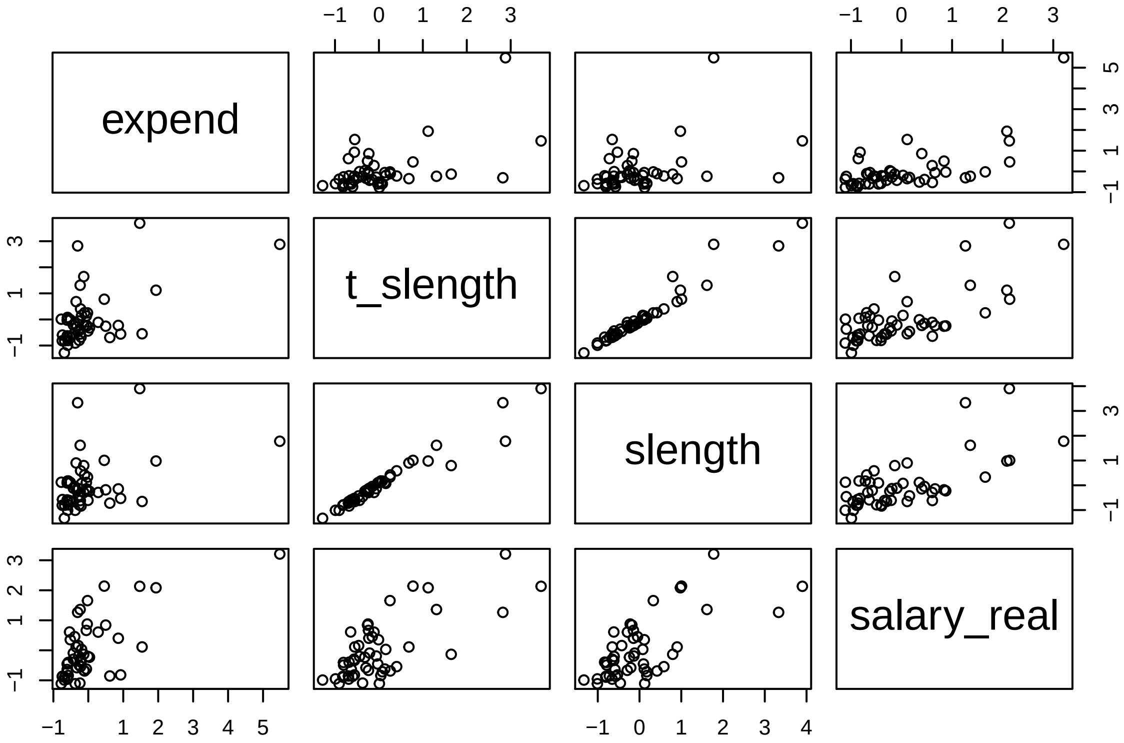
```
##       expend.V1          t_slength.V1          slength.V1    
##  Min.   :-0.772770   Min.   :-1.282138   Min.   :-1.331915  
##  1st Qu.:-0.535853   1st Qu.:-0.599190   1st Qu.:-0.615579  
##  Median :-0.239991   Median :-0.238210   Median :-0.210107  
##  Mean   : 0.000000   Mean   : 0.000000   Mean   : 0.000000  
##  3rd Qu.:-0.022427   3rd Qu.: 0.133236   3rd Qu.: 0.171443  
##  Max.   : 5.478545   Max.   : 3.691295   Max.   : 3.900711  
##    salary_real.V1   
##  Min.   :-1.113266  
##  1st Qu.:-0.714573  
##  Median :-0.296849  
##  Mean   : 0.000000  
##  3rd Qu.: 0.454255  
##  Max.   : 3.206991  
```

Salary and expenditure- the economic variables seem highly skewed. The variables on time, conversely, seem closer to normally distributed.

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.

First, we do so informally by observing pairwise scatterplots.

```
pairs(legprof_sub)
```



Interestingy, there to be some positive relationship between salary and the two types of session lengths. Expenditure seems to be related positively with all three of the other variables, but with a small slope.

In almost all the plots, there is a set of 6-8 outliers towards the upper right of the origin. These outliers do not seem close enough to themselves form a cluster. However, the remaining points do form one close cluster in each plot. Whether this relationship holds in higher dimensions remains to be seen.

Not surprisingly, there is a strong correlation between session length and total session length Let's check this numerically.

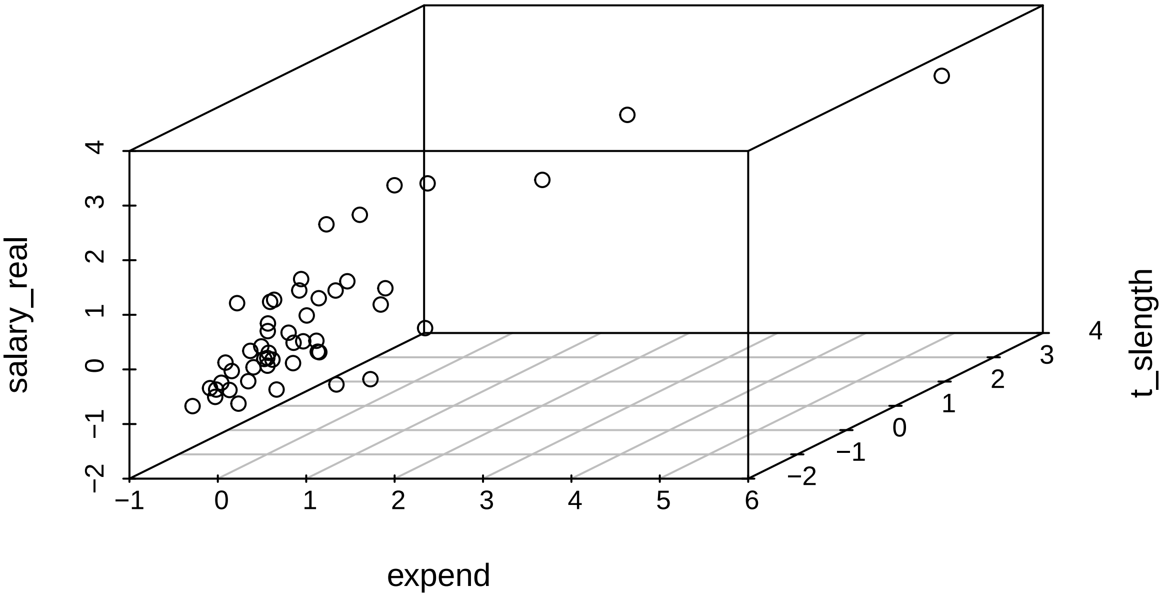
```
cor(legprof_sub$t_slength, legprof_sub$slength)
```

```
##           [,1]
## [1,] 0.9708659
```

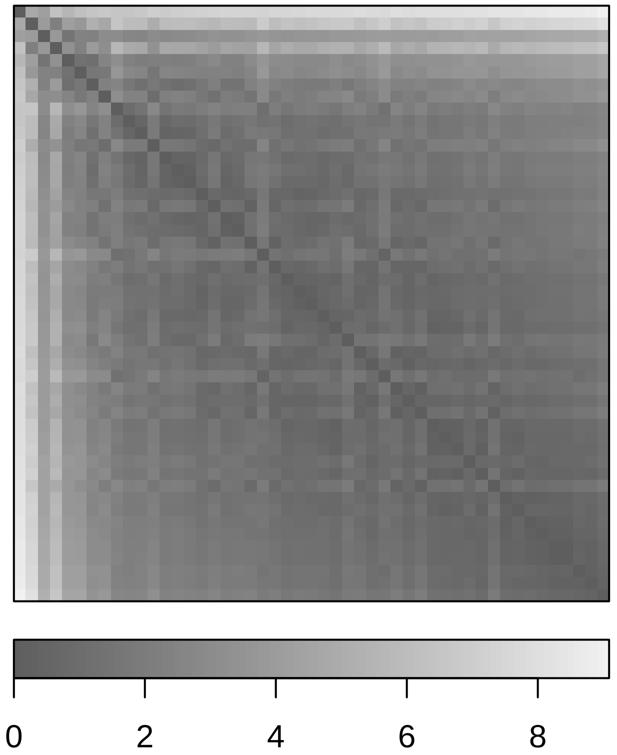
97% correlation suggests a perfectly linear and positive relationship. In this sense, one of the variables can be easily dropped. We will return to this point in later analyses.

If we drop one of them, we will be to check for 3-d scatter plots. So for the sake of experimentation, let's try a 3d version

```
legprof_3d<-legprof_sub %>%
  dplyr::select(expend, t_slength, salary_real)
scatterplot3d(legprof_3d)
```



Just lke in the 2d case, we see a strong clustering in one corner of the plot, with about 8 outliers. Now we turn to more formal methods such as the ODI.



Again, the majority of the data seems to form one cluster, with one less dense cluster in the upper left corner.

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.

```
leg_kmeans <- kmeans(legprof_sub,
                     centers = 2,
                     nstart = 15)
leg_kmeans$size
```

```
## [1] 43  6
```

```
leg_kmeans$cluster
```

```
##  [1] 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 1 1 1 1 1 1 2 1 1 2
## [36] 1 1 2 1 1 1 1 1 1 1 1 1 1 1
```

```
leg_kmeans$centers
```

```
##       expend  t_slength    slength salary_real
## 1 -0.2047966 -0.2930275 -0.2932285  -0.2833616
## 2  1.4677087  2.1000302  2.1014710   2.0307585
```

```
t <-table(leg_kmeans$cluster)
#rownames(t) <- legprof_state$stateabv
#colnames(t)[colnames(t)=="Freq"] <- "Assignment"
```

As per this first classification, we have two clusters- with the majority in the first (43 states) and a minority in the second one.

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(123)
library(mixtools)
gmm<-mvnormalmixEM(as.matrix(legprof_sub),lambda = NULL, mu = NULL, sigma = NULL, k = 2)
```

```
## number of iterations= 16
```

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

We now experiment with another unsupervised method, pam

```
library(cluster)
pam_legprof <- pam(legprof_sub, 2)
pam_legprof
```

```
## Medoids:
##      ID     expend  t_slength    slength salary_real
## [1,] 39 -0.3295059 -0.2949443 -0.2101066  -0.5789619
## [2,] 22  0.4568995  0.7755062  1.0063116   2.1381147
## Clustering vector:
##  [1] 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 1 1 2
## [36] 1 1 2 1 1 1 1 1 1 1 1 1 1 1
## Objective function:
##    build     swap
## 1.188069 1.167453
##
## Available components:
##  [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
##  [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```

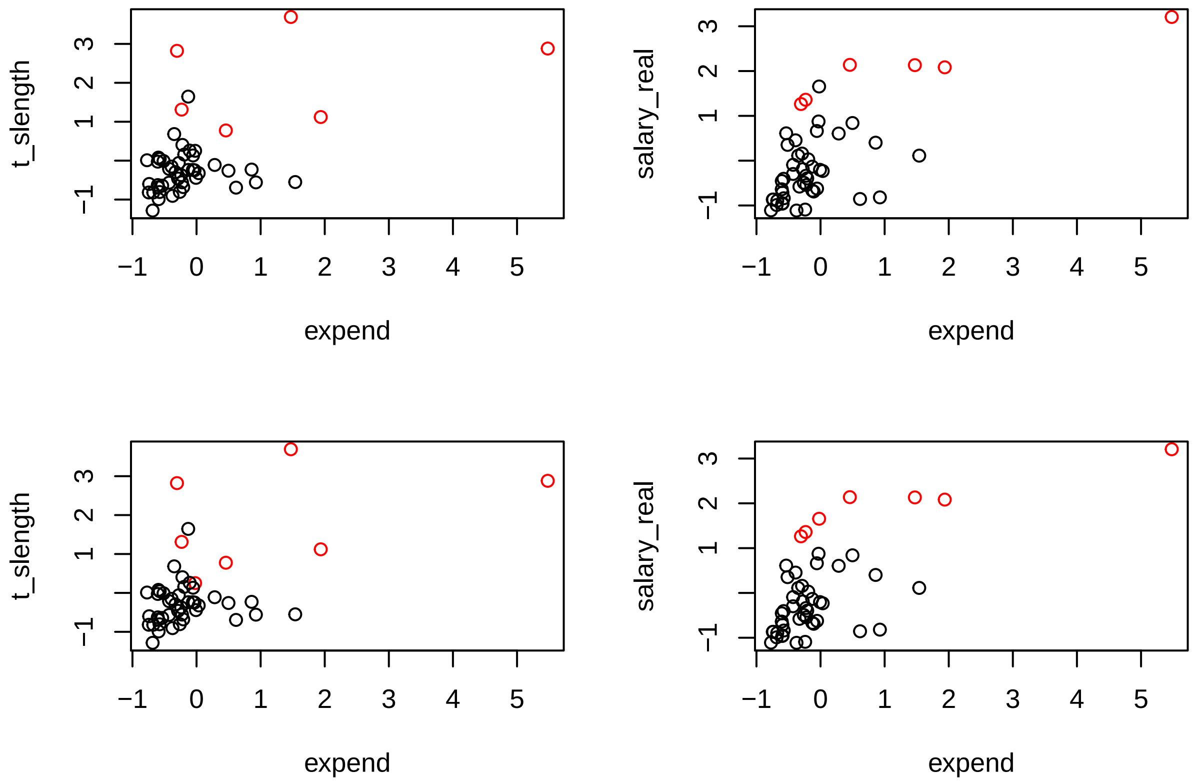
```
pam_legprof$clustering
```

```
##  [1] 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 1 1 2
## [36] 1 1 2 1 1 1 1 1 1 1 1 1 1 1
```

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).

We will compare for 2 scatterplots of interest across all three methods

```
par(mfrow=c(2,2), mar=c(5,4,2,2))
plot(legprof_sub[c(1,2)], col=leg_kmeans$cluster)# Plot to see how Sepal.Length and Sepal.Width data po
plot(legprof_sub[c(1,4)], col=leg_kmeans$cluster)# Plot to see how Sepal.Length and Sepal.Width data po
plot(legprof_sub[c(1,2)], col=pam_legprof$clustering)# Plot to see how Sepal.Length and Sepal.Width dat
plot(legprof_sub[c(1,4)], col=pam_legprof$clustering)# Plot to see how Sepal.Length and Sepal.Width dat
```



We can see that the classification with two clusters is roughly the same across the first and second row of plots (PAM and K Means) respectively.

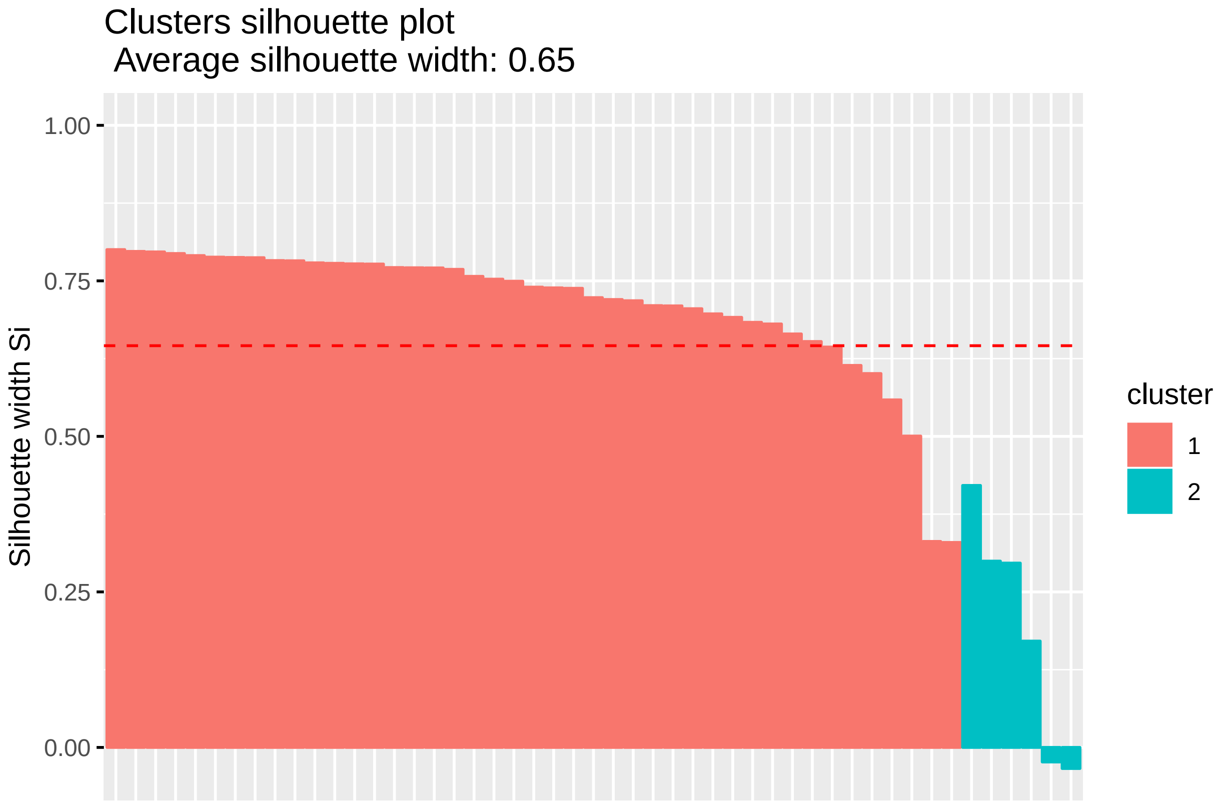
9. Select a single validation strategy (e.g., compactness via min(WSS), average silhouette width, etc.), and calculate for all three algorithms. Display and compare your results for all three algorithms you fit (k-means, GMM, X).

We will be comparing for three strategies across the method of average silhouette width.

First for KMeans

```
km.sil<-silhouette(leg_kmeans$cluster,dist=legprof_dist )
fviz_silhouette(km.sil)
```

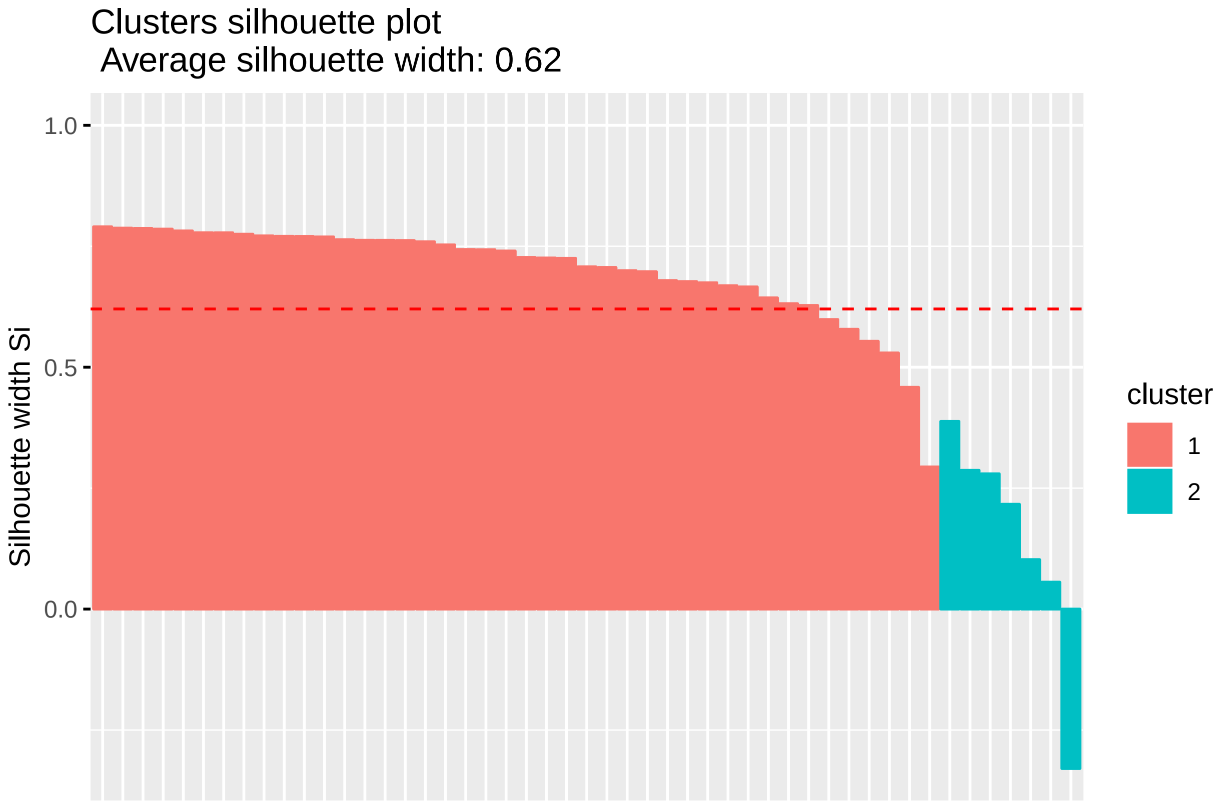
```
##   cluster size ave.sil.width
## 1       1   43          0.71
## 2       2    6          0.19
```



Now the silhouette method for PAM

```
pam.sil<-silhouette(pam_legprof$cluster, dist=legprof_dist)
fviz_silhouette(pam.sil)
```

```
##   cluster size ave.sil.width
## 1       1   42          0.70
## 2       2    7          0.14
```



Based on these three methods, it is clear that the closest fit (with the highest average silhouette is the KMeans approach)

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?