Problem Set 3

Jesús Pacheco

10/21/2019

1.

```
load("~/GitHub/UML-pset3/State Leg Prof Data & Codebook/legprof-components.v1.0.RData")
```

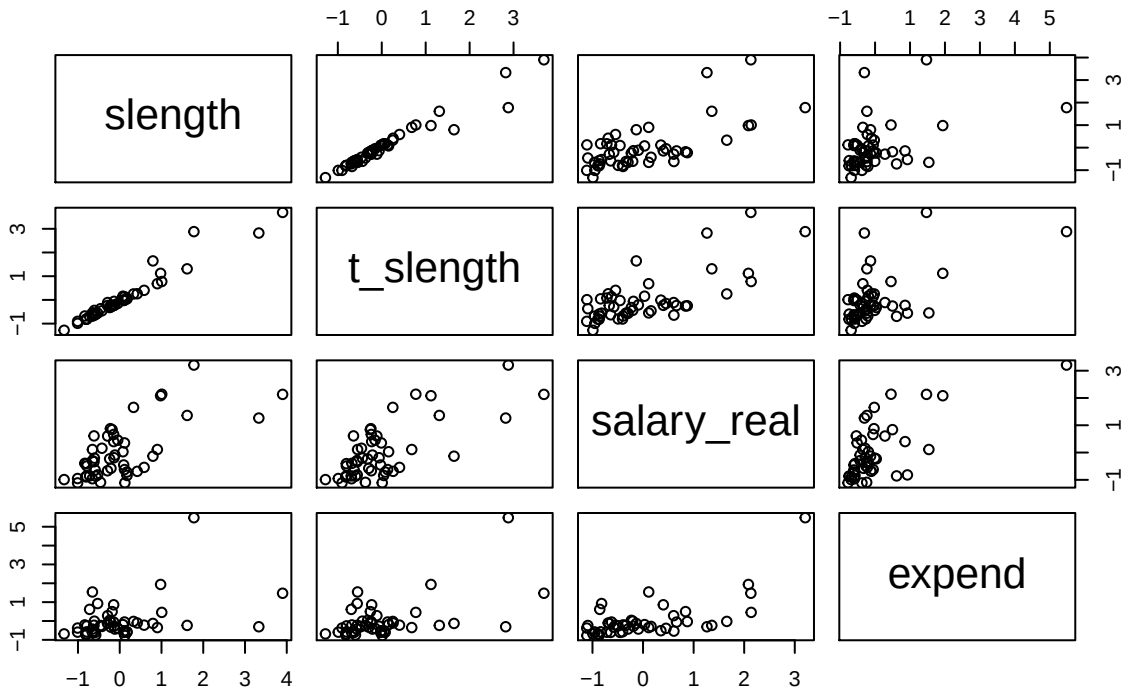
2. The data

```
vars <- x %>% filter(sessid=="2009/10" & t_slength!="NA") %>%
              select(t_slength, slength, salary_real, expend) %>%
              scale()
state_names <- x %>%
              filter(sessid=="2009/10" & t_slength!="NA") %>% select(state, stateabv)
dat <- cbind(state_names, vars)
complete.cases(dat)   #No NAs detected in the data
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [15] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [29] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [43] TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

3. Quick EDA

```
pairs(~slength + t_slength + salary_real + expend, data=dat,
   main="Legistlative Professionalism Scatterplot Matrix")
```



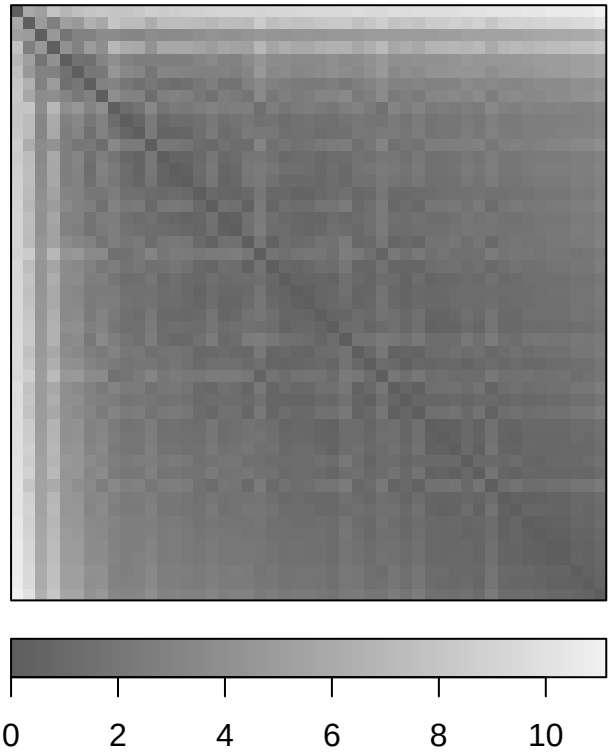
We can see in the scatterplot matrix a potential non-random positive relation in all the interactions between variables. The first two are almost the same variable but we can see that session length seems positively associated with salary and expenditure in local legislatures. Hence, there might be a case of legislature professionalism in these data.

4. Diagnosing clusterability

```
dist_mat<- dist(dat, method = "euclidean")
```

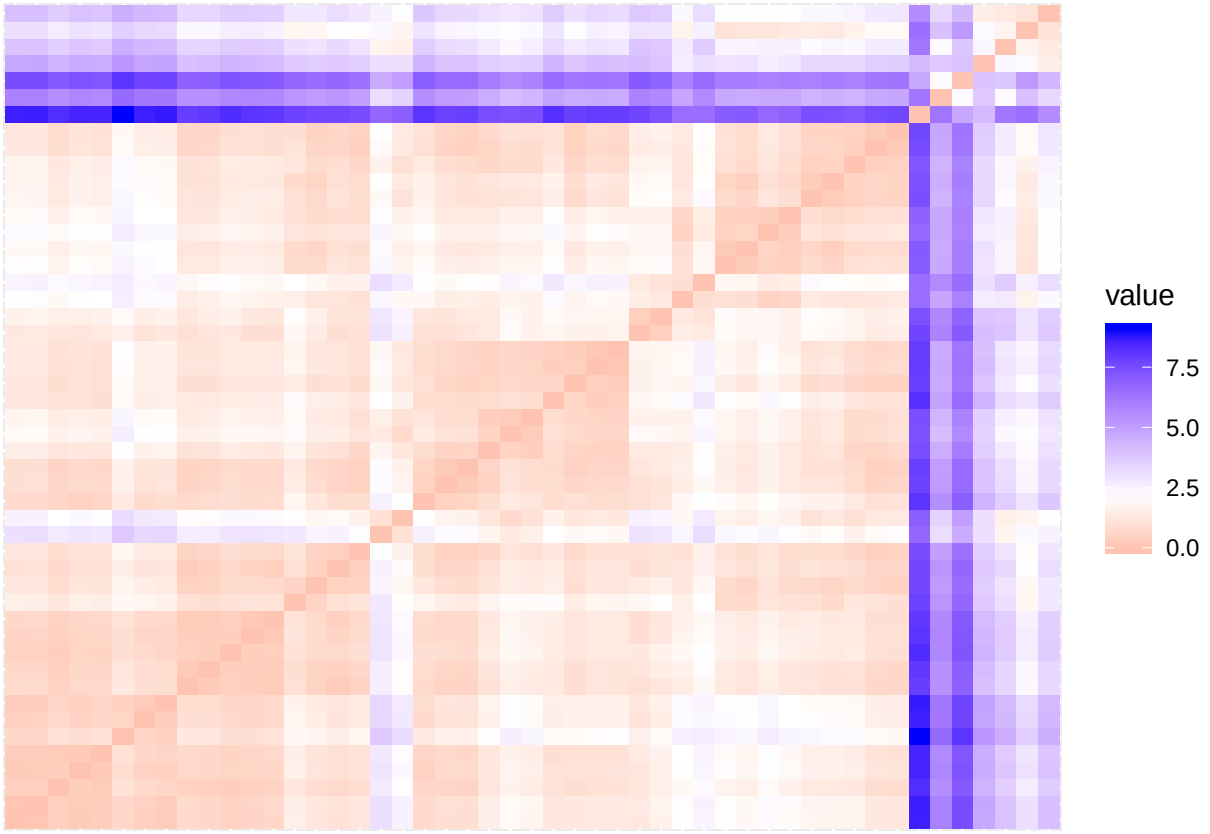
```
## Warning in dist(dat, method = "euclidean"): NAs introduced by coercion
```

```
dissplot(dist_mat)
```

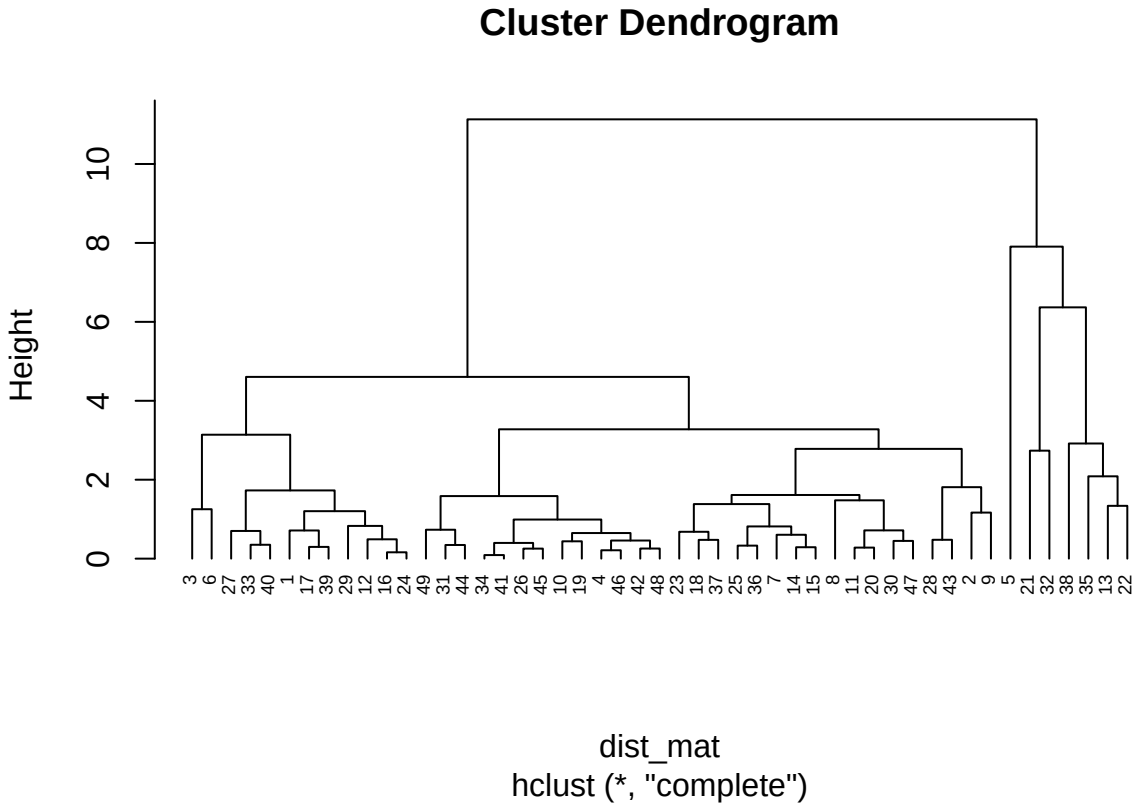



```
get_clust_tendency(data = dat[,3:6], n=10)
```

```
## $hopkins_stat
## [1] 0.2403108
##
## $plot
```



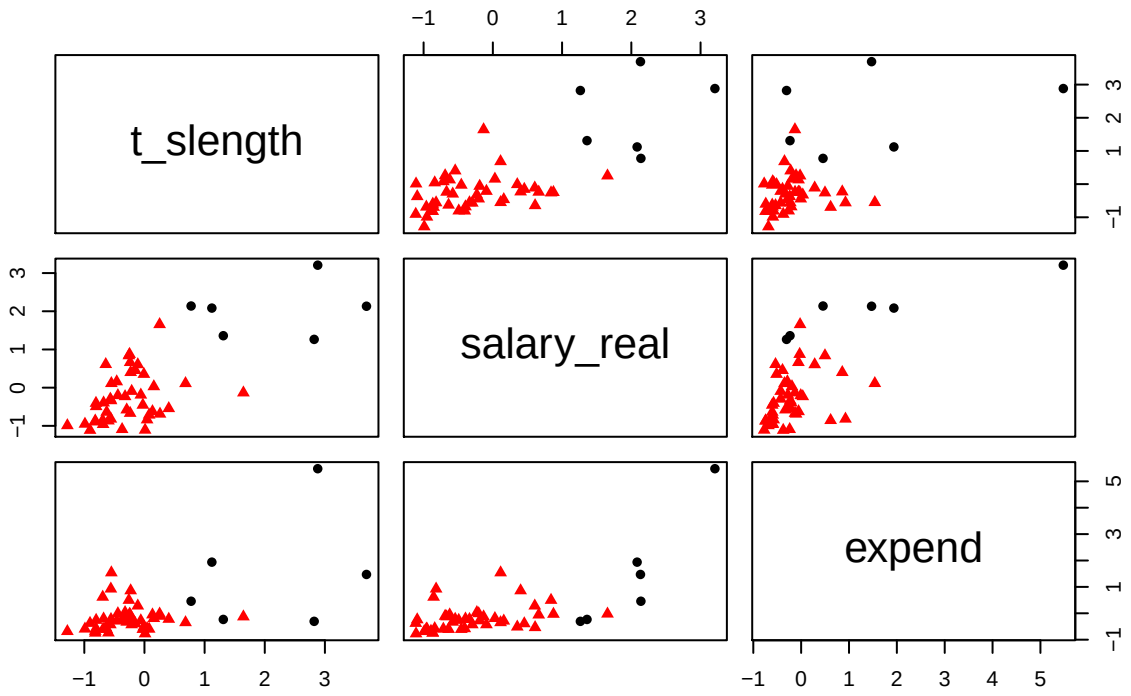

```
#Dendrogram
hc <- hclust(dist_mat, method = "complete"); plot(hc, hang = -1, cex=.6)
```



From the ODI, I do not see a clear pattern along the diagonal that can tell us the clusters are there. Hence, I tried the Hopkins test, which produces a more promising pattern in the visualization but the test is also not promising because the Hopkins statistic is less than 0.5. I tried then, fitting a dendrogram with HCA and I think there might be some case for 2 or potentially 3 clusters where data are condensed.

5. K-means

```
km <- kmeans(dat[,3:6],
             centers = 2,
             nstart = 15)
dat$clust <- as.factor(km$cluster)
pairs(~t_slength + salary_real + expend, data=dat, col=as.factor(dat$clust),
      pch = c(16, 17)[as.numeric(dat$clust)],
      main="Legistlative Professionalism Scatterplot Matrix")
```




```
dat %>% group_by(clust) %>% summarise(mean_slength = mean(slength), mean_salary = mean(salary_real))
```

```
## # A tibble: 2 x 3
##   clust mean_slength mean_salary
##   <fct>        <dbl>       <dbl>
## 1 1             2.10        2.03
## 2 2            -0.293      -0.283
```

It seems like the algorithm is clustering the observations that are far away in the distribution together (legislatures with greater number of sessions). This might be a good sign because we see in all the scatterplots the clusters close together in the distribution, but I wonder if we are using all the granularity/information in the left side of the distribution since the algorithm might be stopping when outliers are clustered together.

6. Gaussian mixture model

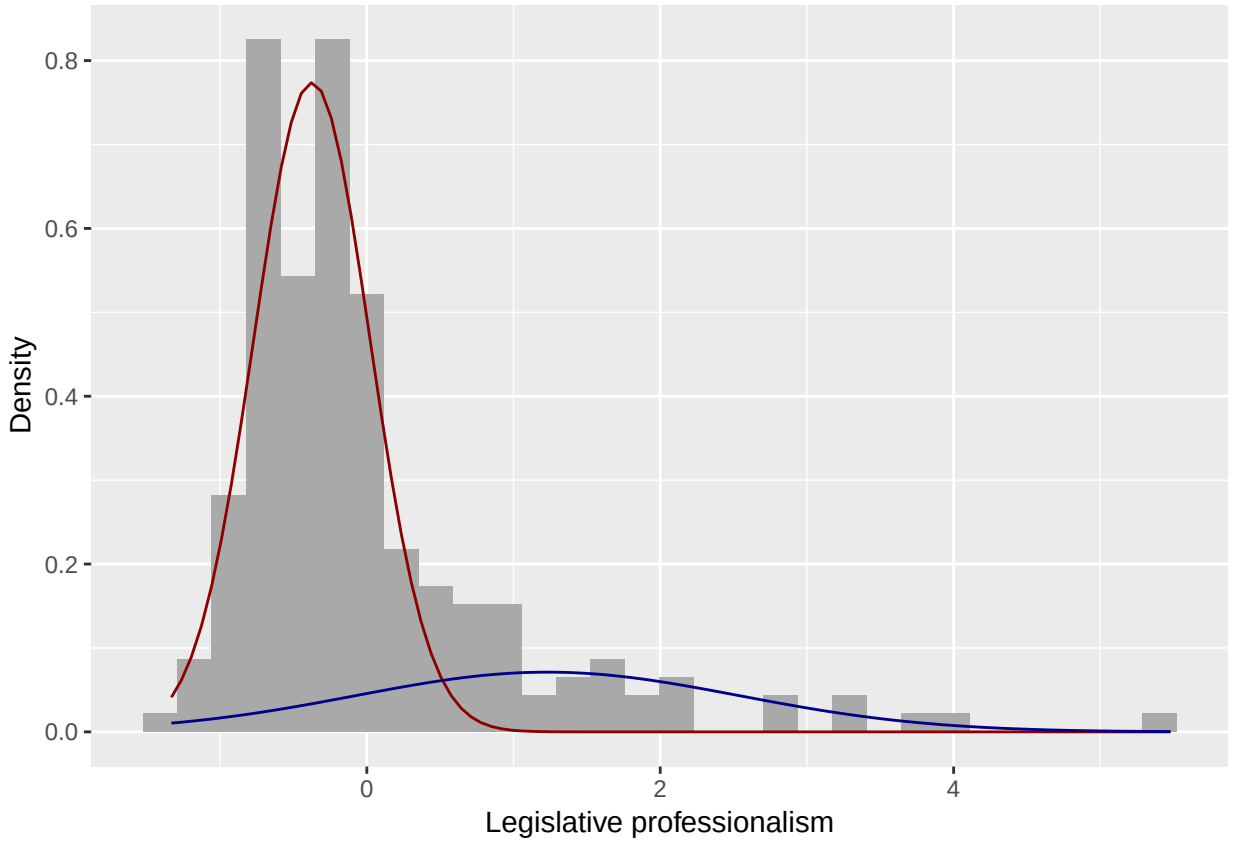
```
set.seed(123)
gmm <- normalmixEM(as.matrix(dat[,3:6]),
            k = 2)
```

```
## number of iterations= 36
```

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

```
  ylab("Density")
```

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




```
post_gmm <- as.data.frame(cbind(x = gmm$x, gmm$posterior))
post_gmm$clust <- ifelse(post_gmm$comp.1>post_gmm$comp.2, 1,2)
post_gmm %>% ggplot(aes(x, clust)) + geom_point()
```

As we can see from the graph, the Gaussiam mixture model is characterizing the two clusters in this space in the following way: the observations that are the right of the distribution (values close to the median) in one cluster, and values that are far from the median (state legislatures that have outlying values of expenses, salary and numbers of sessions) as a second cluster which is more spread.

```
gmm$mu
```

```
## [1] -0.3745312  1.2310883
```

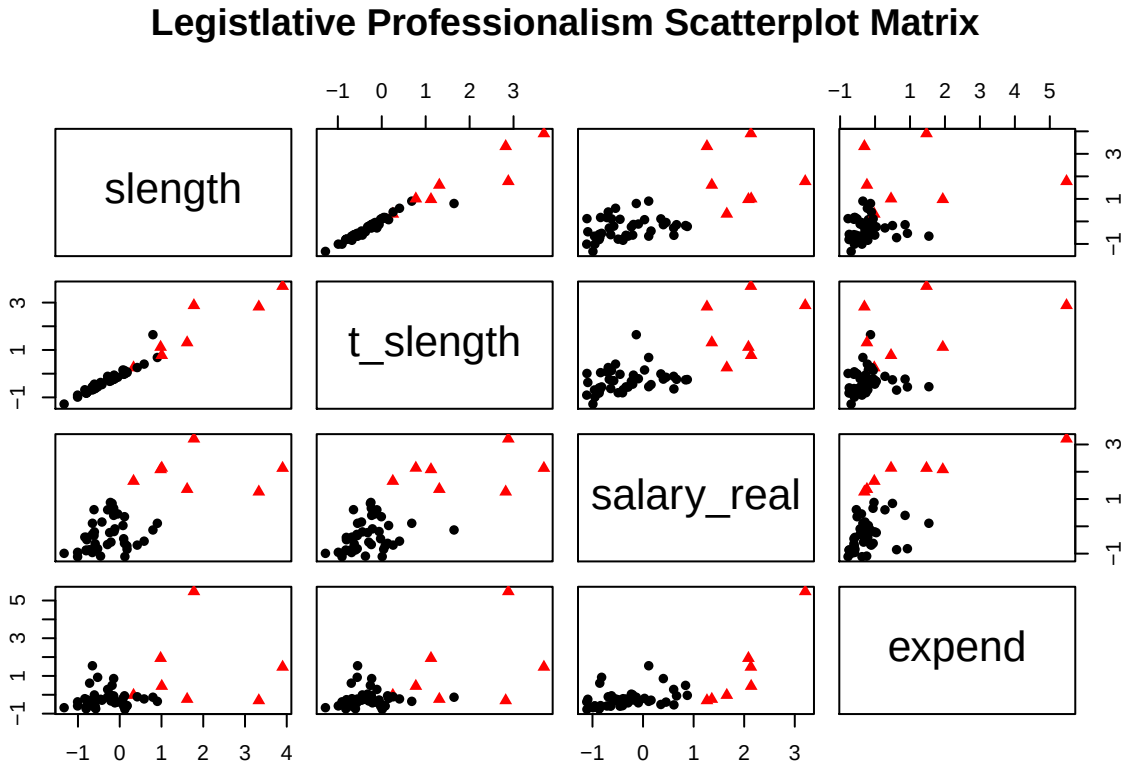
```
gmm$sigma
```

```
## [1] 0.3953362 1.3073380
```

We can see in this result the computed central tendendy of the distribution of the two components (clusters). The first one is to the right but with significantly more variation than the second one. Althought there is not a lot of interpretation for these numbers, we can see numerically what we see in the graph: the first soft partitioning is imprecise because is clustering the outliers together.

7. One more partitioning technique

```
set.seed(123)
pm <- pam(dat[,3:6],
              k = 2)
dat$clust_pam<- as.factor(pm$cluster)
pairs(~slength + t_slength + salary_real + expend, data=dat,
                        col=as.factor(dat$clust_pam), pch = c(16, 17)[as.numeric(dat$clust_pam)],
   main="Legistlative Professionalism Scatterplot Matrix")
```




```
dat %>% group_by(clust) %>% summarise(mean_slength = mean(slength), mean_salary = mean(salary_real))
```

```
## # A tibble: 2 x 3
##   clust mean_slength mean_salary
##   <fct>        <dbl>       <dbl>
## 1 1             2.10        2.03
## 2 2            -0.293      -0.283
```

```
length(which(dat$clust!=dat$clust_pam))
```

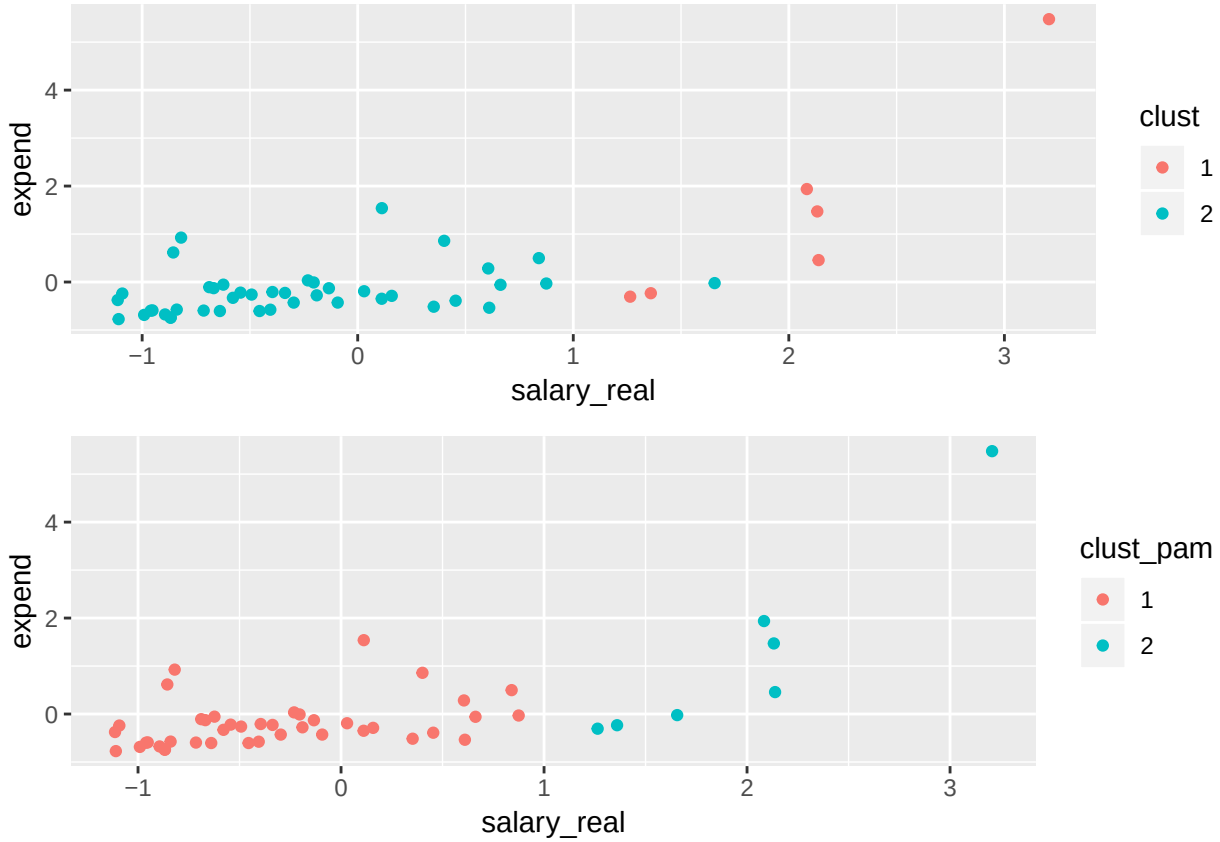
```
## [1] 48
```

As expected the classification with PAM is pretty similar to the k-means one, the data are generating clusters depending on outliers observations. Actually in the data we can see only one difference with the k-means model.

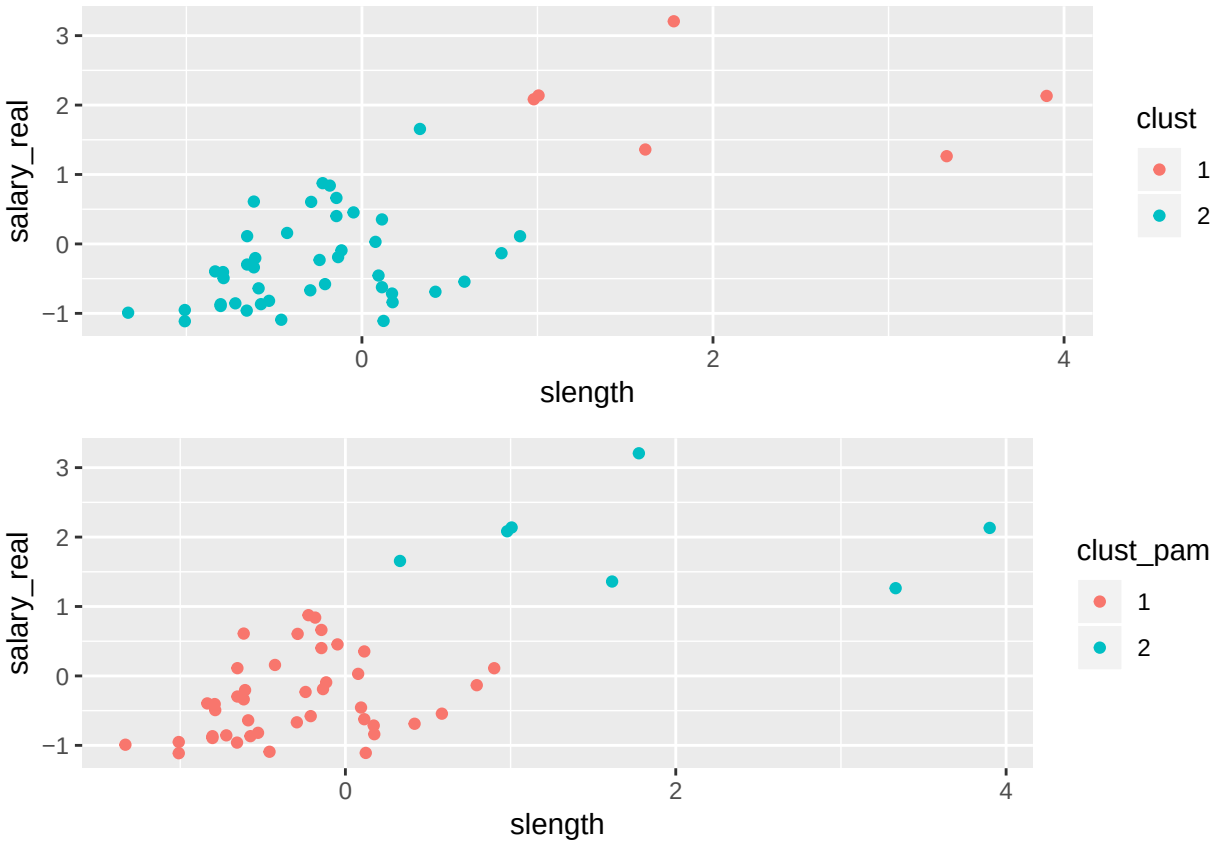
8.

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

```
clust1 <- dat %>%  ggplot(aes(x=salary_real, y=expend, col=clust)) + geom_point() +
  labs(main="Clustering with k-means")
clust2 <- dat %>%  ggplot(aes(x=salary_real, y=expend, col=clust_pam)) + geom_point() +
  labs(main="Clustering with PAM")
grid.arrange(clust1, clust2, nrow=2)
```




```
clust3 <- dat %>%  ggplot(aes(x=slength, y=salary_real, col=clust)) + geom_point() +
  labs(main="Clustering with k-means")
clust4 <- dat %>%  ggplot(aes(x=slength, y=salary_real, col=clust_pam)) + geom_point() +
  labs(main="Clustering with PAM")
grid.arrange(clust3, clust4, nrow=2)
```



As mentioned before, we can see that the clusters are clear and pretty similar in both methods. In the first graph we can see that PAM might be doing a better job with expenditure and salary (one blue dot alone in the right side of the distribution).

9.

```
connectivity(dist(dat[,3:5]), dat$clust)
```

```
## [1] 7.554762
```

```
connectivity(dist(dat[,3:5]), dat$clust_pam)
```

```
## [1] 6.48373
```

```
#connectivity(dist(dat[,3:5]), dat$component)
```

As we saw in the graph in last question, PAM did a better job with the clustering using the connectivity validation measurement. Since connectivity measures compactness within cluster, we expect algorithms with lower connectivity scores to be a better fit.

10. The validation output.

```
internal <- clValid(dat[,3:6],nClust = 2:10,
                    clMethods = c("kmeans", "pam"),
                    validation = "internal")
```

```
## Warning in clValid(dat[, 3:6], nClust = 2:10, clMethods = c("kmeans",
```

```
## "pam"), : rownames for data not specified, using 1:nrow(data)
```

```
summary(internal)
```

```
##
## Clustering Methods:
##  kmeans pam
##
## Cluster sizes:
##  2 3 4 5 6 7 8 9 10
##
## Validation Measures:
##                             2       3       4       5       6       7       8       9      10
##
## kmeans Connectivity   8.4460 10.8960 16.1885 28.7437 30.7437 37.5266 39.4552 40.8694 45.6623
##        Dunn           0.1735  0.2581  0.2562  0.1090  0.1090  0.1108  0.1260  0.1324  0.1386
##        Silhouette     0.6458  0.6131  0.4932  0.3042  0.2858  0.2750  0.3131  0.3307  0.3288
## pam    Connectivity   7.9071 21.2952 25.4798 26.3464 35.7008 42.7266 49.2762 51.2762 53.8071
##        Dunn           0.1673  0.0324  0.0332  0.0670  0.0731  0.0991  0.1019  0.1019  0.1130
##        Silhouette     0.6204  0.2530  0.2673  0.2841  0.2993  0.2682  0.2564  0.2385  0.2415
##
## Optimal Scores:
##
##              Score  Method Clusters
## Connectivity 7.9071 pam    2
## Dunn         0.2581 kmeans 3
## Silhouette   0.6458 kmeans 2
```

What can you take away from the fit? The main takeaway for me is that there is no unique process to do when trying to find clusterability. In this internal validation process we can see that two clusters seem ideal most of the time, hence we can think of two types of local legistature professionalism in the United States.

Which approach is optimal? And optimal at what value of k? I would choose the PAM method with k=2. The cluster are more compact with this method and the salary and expenditure relation was better clustered with the PAM as shown in previous graphs.

What are reasons you could imagine selecting a technically "sub-optimal" partitioning method, regardless of the validation statistics? I could imagine that in some cases when we have strong priors about k, or we need to assign the data to a certain amount of clusters, we could still go with a k that is technically sub-optimal.