CMAP PS1

Alec MacMillen

10/22/2019

Question 1 - Data Loading

Load the state legislative professionalism data.

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

Question 2 - Data Munging

Munge the data: 1. Select only continuous features, 2. Restrict data to 2009-10 legislative session, 3. Omit missing values, 4. Standardize input features, 5. Other steps necessary to make the data in workable form.

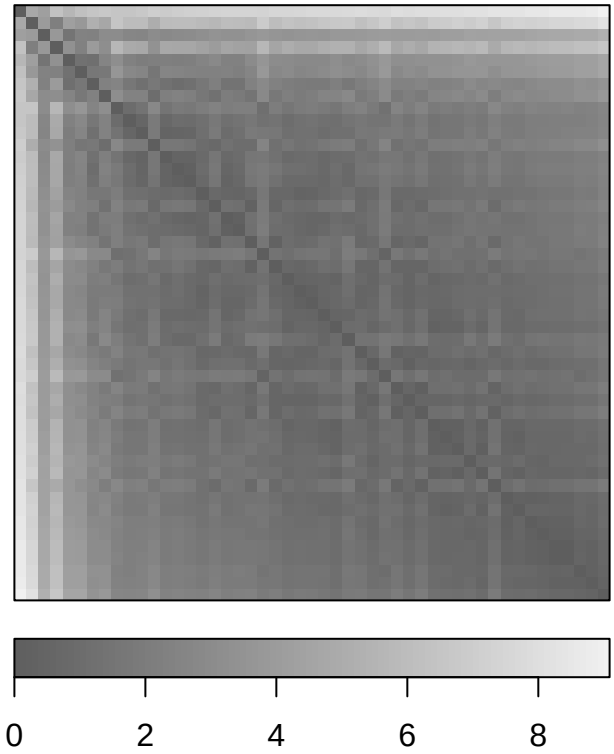
```
legprof <- x %>%
  select(state:expend) %>%
  filter(sessid == "2009/10", complete.cases(.))

rownames(legprof) <- legprof$state

legprof_numeric <- legprof %>%
  select(-c(state, sessid))

legprof_scale <- legprof_numeric %>%
  scale()
```

Question 3 - Diagnosing Clusterability

```
# Hopkins statistic
h_stat <- clustertend::hopkins(legprof_numeric, n = nrow(legprof)-1)
h_stat
```

```
## $H
## [1] 0.1613968
```

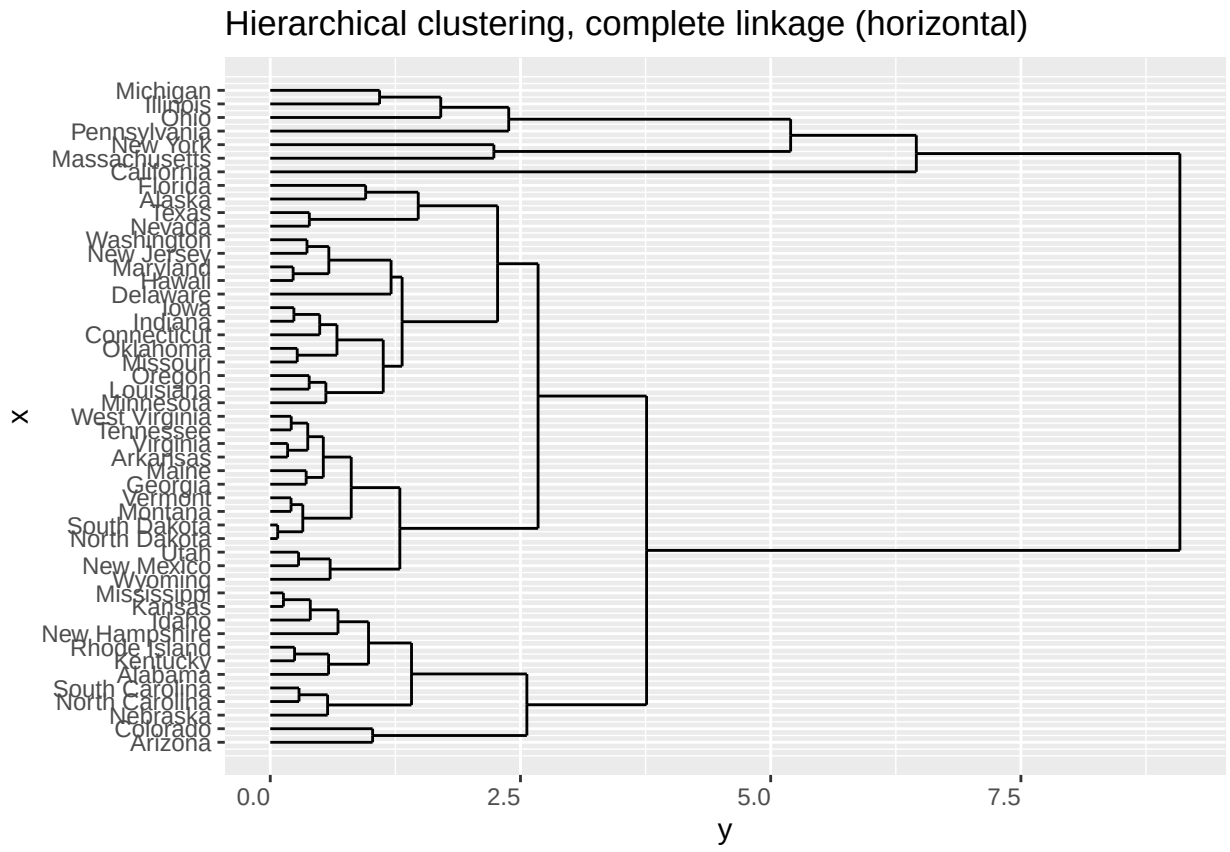
```
# VAT ODI plot
# Create distance matrix
lpdist <- legprof_scale %>% dist()
seriation::dissplot(lpdist)
```



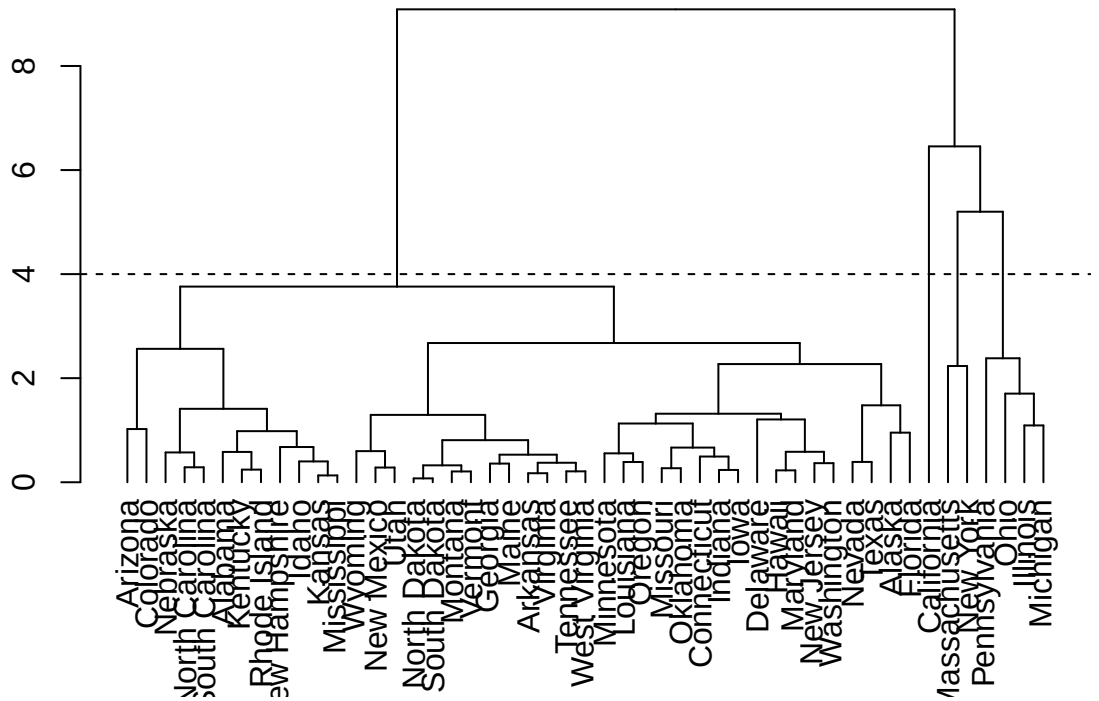
The Hopkins statistic is 0.156, which is far from the $H = 0.5$ threshold, suggesting that the data is not *highly* clusterable, but is probably not generated by a completely random process - more likely a uniform one. This is not entirely unexpected given that we're attempting to cluster only 49 observations. The VAT ODI plot corroborates this: while there are not several well-formed black blocks representing clusters, the plot is more ordered than one that would result from a truly random dataset. This suggests that a cluster analysis could yield some useful and interesting results.

Question 4 - Hierarchical clustering algorithm

```
hc_complete <- lpdist %>% hclust(method = "complete") %>% as.dendrogram

ggdendrogram(hc_complete, rotate = TRUE, theme_dendro = FALSE, title = "Hierarchical clustering, comple
  ggtitle("Hierarchical clustering, complete linkage (horizontal)")
```




```
hc_complete %>%
  plot() %>%
  title(main = "Hierarchical clustering, complete linkage", line = 1)
abline(h = 4, lty = 2)
```



An agglomerative hierarchical clustering algorithm using complete linkage appears to first cluster between what could be called a "highly professional" set of legislatures (CA, MA, NY, PA, OH, IL, MI) and all other remaining legislatures.

```
legprof %>%
  mutate(prof_check = ifelse(state %in% c("California", "Massachusetts", "New York",
                                          "Pennsylvania", "Ohio", "Illinois", "Michigan"), 1, 0)) %>%
  group_by(prof_check) %>%
  summarize(med_sal = median(salary_real),
            med_exp = median(expend))
```

```
## # A tibble: 2 x 3
##   prof_check med_sal med_exp
##        <dbl>   <dbl>   <dbl>
## 1          0    33.7    498.
## 2          1   158.    1143.
```

A cursory comparison of these two clustered groups reveal that the smaller cluster of highly professional legislatures have a much higher median salary and median expenditures per legislator, as we might expect. The dendrogram appears to indicate that a large cluster split occurs when moving from 4 to 5 clusters (where the dashed horizontal line is), so let's examine why.

```
cuts <- dendextend::cutree(hc_complete, k = c(4,5))

### Or, a simple matrix of assignments by iteration
```

```
table(`4 Clusters` = cuts[,1],
      `5 Clusters` = cuts[,2])
```

```
##           5 Clusters
## 4 Clusters  1  2  3  4  5
##          1 12 30  0  0  0
##          2  0  0  1  0  0
##          3  0  0  0  4  0
##          4  0  0  0  0  2
```

Fully 30 states move from cluster 1 to cluster 2 when 5 clusters are used instead of 4.

```
hclust_5 <- tibble(state = rownames(legprof), clust5 = cuts[,2])
legprof %>%
  left_join(hclust_5, by = "state") %>%
  group_by(clust5) %>%
  summarize(med_sal = median(salary_real),
            med_exp = median(expend),
            med_tslength = median(t_slength),
            med_slength = median(slength))
```

```
## Warning: Column `state` has different attributes on LHS and RHS of join
```

```
## # A tibble: 5 x 5
##   clust5 med_sal med_exp med_tslength med_slength
##    <int>   <dbl>   <dbl>        <dbl>       <dbl>
## 1      1    23.1    496.         153.        149.
## 2      2    42.0    498.         101.         94.3
## 3      3   213.    5523.         390         270
## 4      4   147.     934.         228.        212
## 5      5   139.    1253.         422.        406.
```

When 5, rather than 4, clusters are used in hierarchical agglomerative clustering, a group of 30 states is split from cluster 1 into cluster 2 that appears to have a higher median salary and shorter session length than the states remaining in cluster 1.

Question 5 - K-means

```
set.seed(678)
kmeans_2 <- kmeans(legprof_scale, centers = 2, nstart = 15)
str(kmeans_2)
```

```
## List of 9
##  $ cluster     : Named int [1:49] 2 2 2 2 1 2 2 2 2 2 ...
##   ..- attr(*, "names")= chr [1:49] "Alabama" "Alaska" "Arizona" "Arkansas" ...
##  $ centers     : num [1:2, 1:4] 2.1 -0.293 2.101 -0.293 2.031 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:4] "t_slength" "slength" "salary_real" "expend"
```

```
##  $ totss       : num 192
##  $ withinss    : num [1:2] 40.4 48.4
##  $ tot.withinss: num 88.7
##  $ betweenss   : num 103
##  $ size        : int [1:2] 6 43
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
kmeans_2_df <- tibble(state = rownames(legprof), k2 = kmeans_2$cluster)
kmeans_2_df %>% filter(k2 == 1) %>% select(state)
```

```
## # A tibble: 6 x 1
##   state
##   <chr>
## 1 California
## 2 Massachusetts
## 3 Michigan
## 4 New York
## 5 Ohio
## 6 Pennsylvania
```

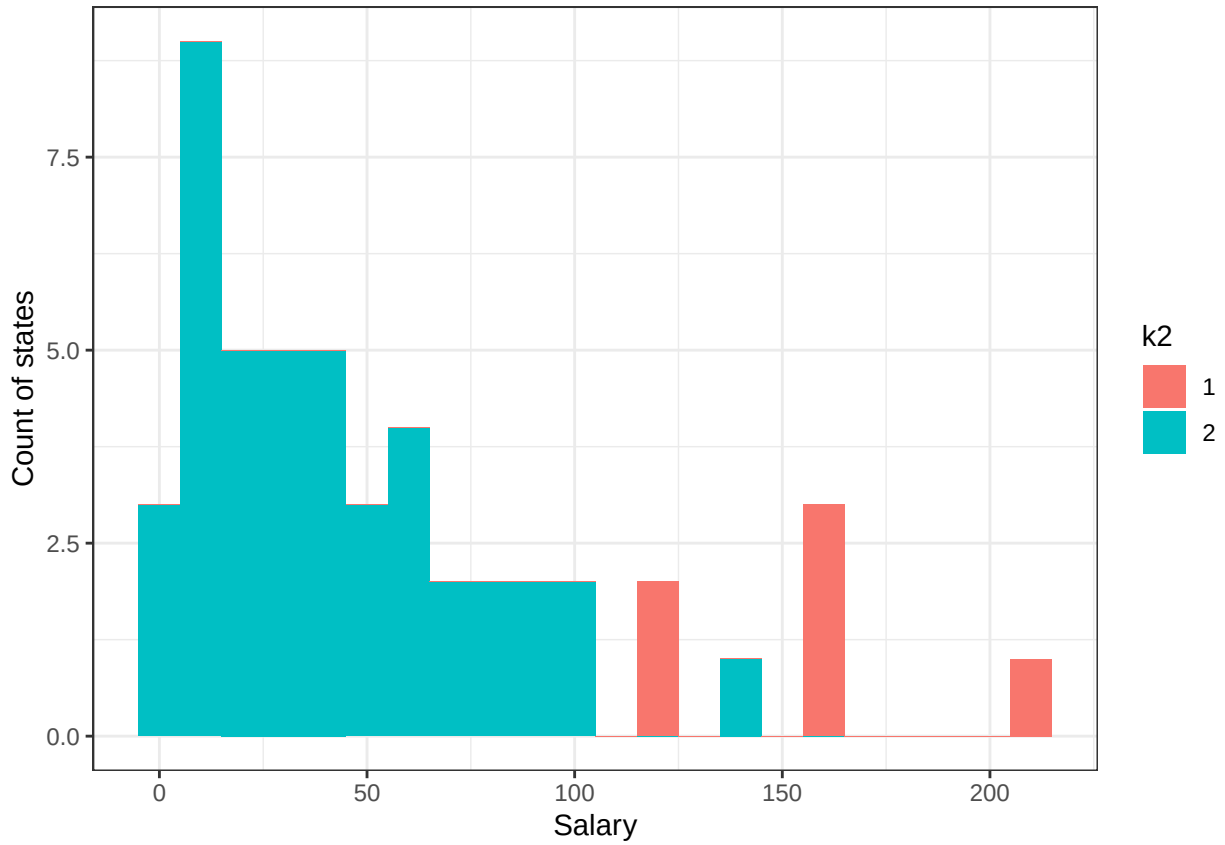
```
legprof %>%
  left_join(kmeans_2_df, by = "state") %>%
  group_by(k2) %>%
  summarize(med_sal = median(salary_real),
            med_exp = median(expend))
```

```
## Warning: Column `state` has different attributes on LHS and RHS of join
```

```
## # A tibble: 2 x 3
##      k2 med_sal med_exp
##   <int>   <dbl>   <dbl>
## 1     1   159.    1585.
## 2     2    35.0    503.
```

```
legprof %>%
  left_join(kmeans_2_df, by = "state") %>%
  mutate(k2 = as.factor(k2)) %>%
  ggplot(aes(salary_real, fill = k2)) +
  geom_histogram(binwidth = 10) +
  labs(x = "Salary", y = "Count of states") +
  theme_bw()
```

```
## Warning: Column `state` has different attributes on LHS and RHS of join
```



The k-means clustering algorithm appears to perform a similar clustering to the hierarchical algorithm - a small group of "highly professional" legislatures with much higher median salary and expenditures per legislature (in this case, California, Massachusetts, Michigan, New York, Ohio and Pennsylvania - the same group of states as before but without Illinois) are separated from the rest of the states into their own cluster immediately. However, a quick histogram of states by salary with cluster colors applied shows that we may have misclassified a state into the second, "less professional" cluster when its salary might suggest that it should be in the "more professional" cluster. This is one indication that 2 clusters might not be enough to adequately capture all the variation in the dataset.

Question 6 - Gaussian mixture model

Because our experience with previous model fits suggests that clusters are strongly distinguished by a state legislature's real salary, let's fit a Gaussian mixture model

```
# Let's cluster specifically on real salary
gmm1 <- mixtools::normalmixEM(legprof[,5], k = 2)
```

```
## number of iterations= 68
```

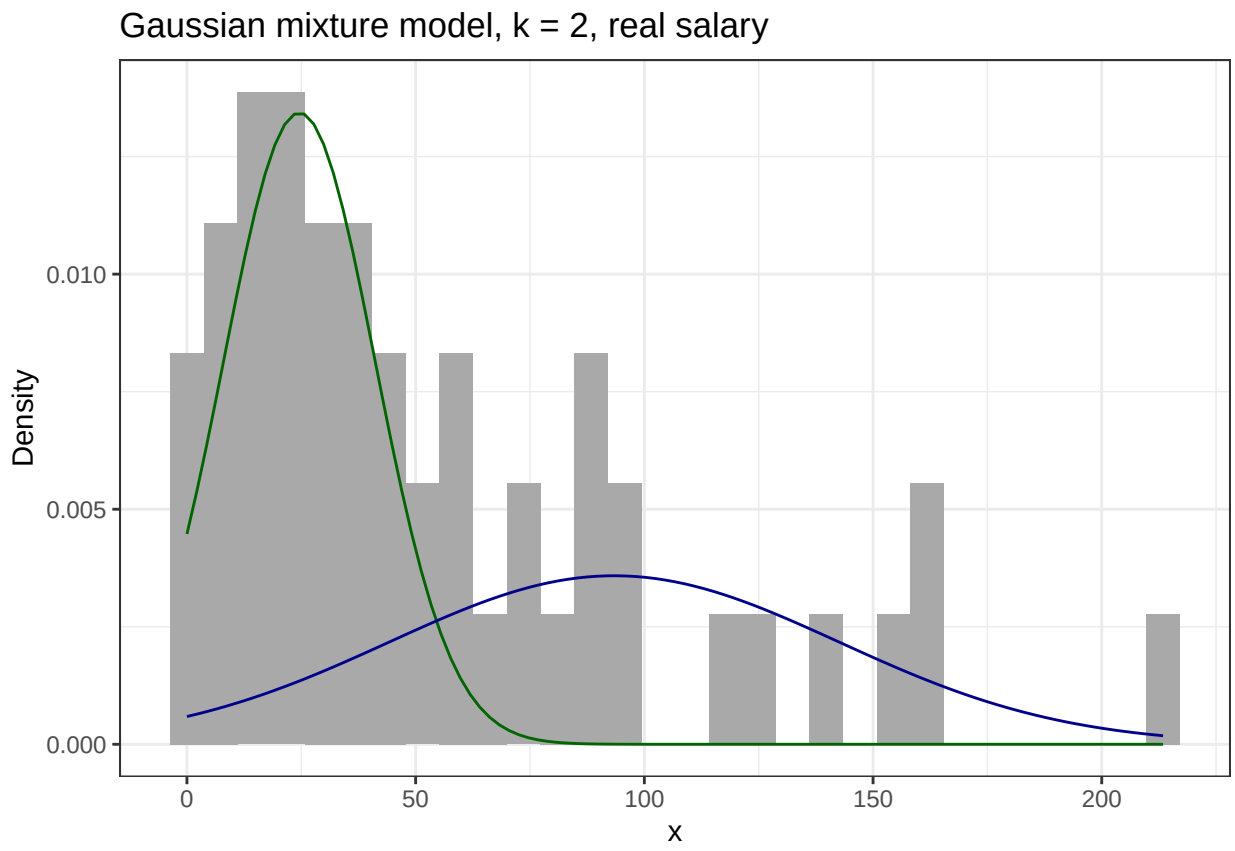
```
str(gmm1)
```

```
## List of 9
##  $ x         : num [1:49] 1.05 74.81 48.39 30.67 213.41 ...
##  $ lambda    : num [1:2] 0.558 0.442
```

```
##  $ mu        : num [1:2] 24.6 93.4
##  $ sigma     : num [1:2] 16.6 49.2
##  $ loglik    : num -250
##  $ posterior : num [1:49, 1:2] 8.89e-01 3.92e-02 6.70e-01 8.88e-01 4.76e-27 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:2] "comp.1" "comp.2"
##  $ all.loglik: num [1:69] -279 -257 -254 -253 -252 ...
##  $ restarts  : num 0
##  $ ft        : chr "normalmixEM"
##  - attr(*, "class")= chr "mixEM"
```

```r
ggplot(data.frame(x = gmm1$x)) +
  geom_histogram(aes(x, ..density..), fill = "darkgray") +
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm1$mu[1], gmm1$sigma[1], lam = gmm1$lambda[1]),
                colour = "darkgreen") +
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm1$mu[2], gmm1$sigma[2], lam = gmm1$lambda[2]),
                colour = "darkblue") +
  ylab("Density") +
  theme_bw() +
  ggtitle("Gaussian mixture model, k = 2, real salary")
```

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



Fitting a Gaussian mixture model with 2 clusters on real salary results in one fairly tight distribution and another that is much more spread - it could be possible that the high outlier (California) is skewing the distribution, or it could be the case that a bimodal distribution is not the best approximation of how the data are generated.

One important note is that the real salary for California legislators dwarfs the salary of legislators in every other state - let's try dropping California and seeing whether that impacts the model.

```
# Now let's try dropping California and repeating the process
legprof_noca <- legprof[-c(5),]
gmm2 <- mixtools::normalmixEM(legprof_noca[,5], k = 2)
```

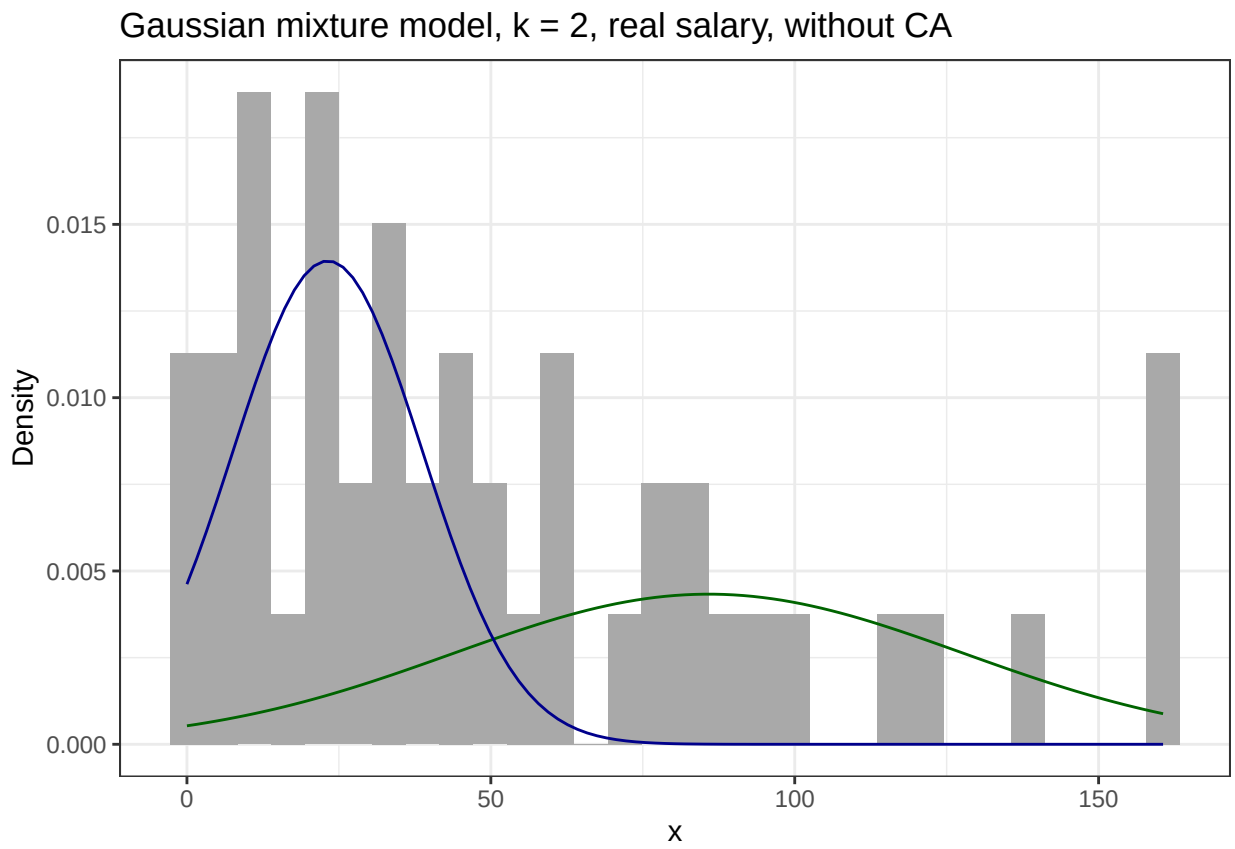
```
## number of iterations= 67
```

```
str(gmm2)
```

```
## List of 9
##  $ x         : num [1:48] 1.05 74.81 48.39 30.67 60.49 ...
##  $ lambda    : num [1:2] 0.455 0.545
##  $ mu        : num [1:2] 85.8 23.2
##  $ sigma     : num [1:2] 41.9 15.6
##  $ loglik    : num -241
##  $ posterior : num [1:48, 1:2] 0.0992 0.9864 0.436 0.128 0.8201 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:2] "comp.1" "comp.2"
##  $ all.loglik: num [1:68] -287 -253 -251 -250 -250 ...
##  $ restarts  : num 0
##  $ ft        : chr "normalmixEM"
##  - attr(*, "class")= chr "mixEM"
```

```
ggplot(data.frame(x = gmm2$x)) +
  geom_histogram(aes(x, ..density..), fill = "darkgray") +
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm2$mu[1], gmm2$sigma[1], lam = gmm2$lambda[1]),
                colour = "darkgreen") +
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm2$mu[2], gmm2$sigma[2], lam = gmm2$lambda[2]),
                colour = "darkblue") +
  ylab("Density") +
  theme_bw() +
  ggtitle("Gaussian mixture model, k = 2, real salary, without CA")
```

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



Doesn't seem to make that much of a difference - we still have one fairly tight distribution and one fairly spread.

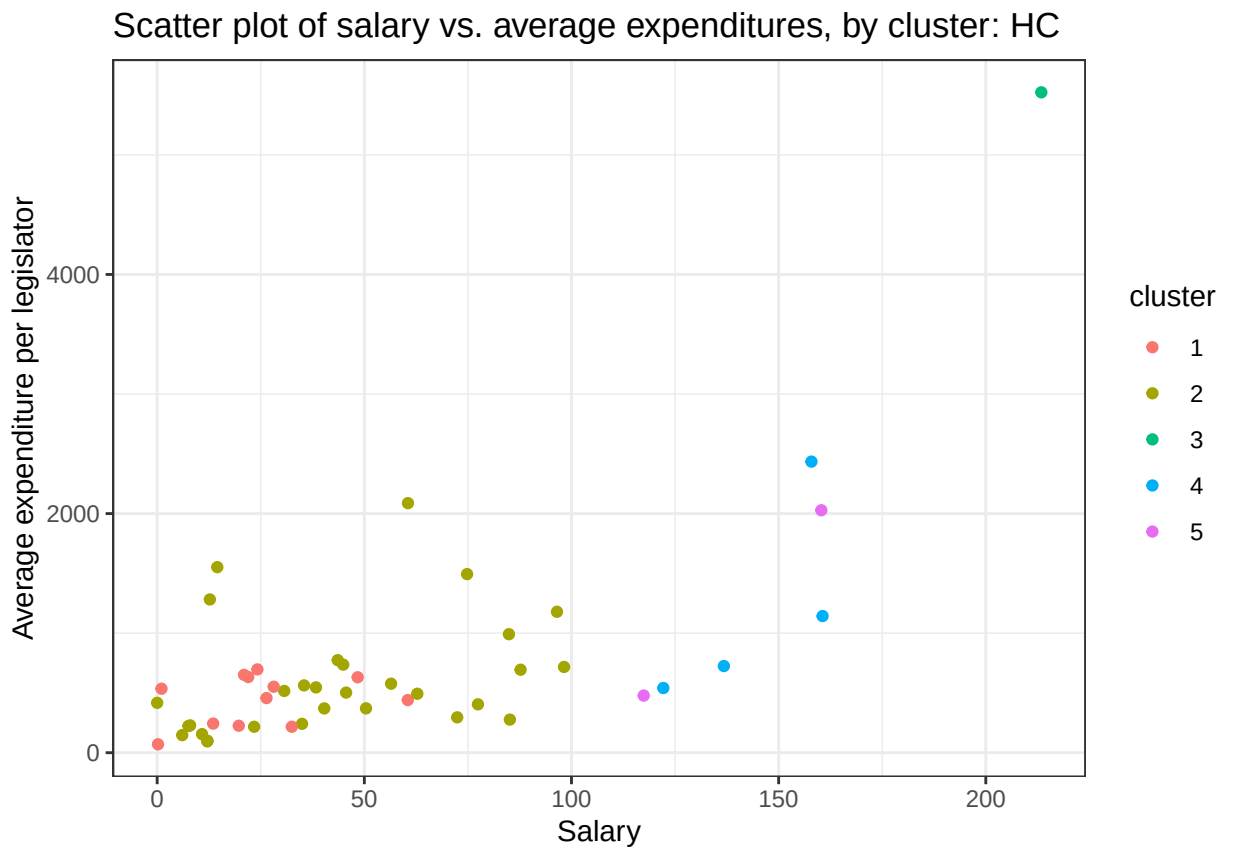
Question 7: Comparing output

In addition to the plots presented above, let's look at scatter plots of legislator salaries against average expenditure per legislator and how these points fall into different clusters, by method.

```
q7_plot_hc5 <- legprof %>%
  left_join(hclust_5, by = "state") %>%
  mutate(cluster = as.factor(clust5)) %>%
  ggplot(aes(x = salary_real, y = expend, color = cluster)) +
  geom_point() +
  labs(x = "Salary", y = "Average expenditure per legislator") +
  theme_bw() +
  ggtitle("Scatter plot of salary vs. average expenditures, by cluster: HC")
```

```
## Warning: Column `state` has different attributes on LHS and RHS of join
```

```
q7_plot_hc5
```

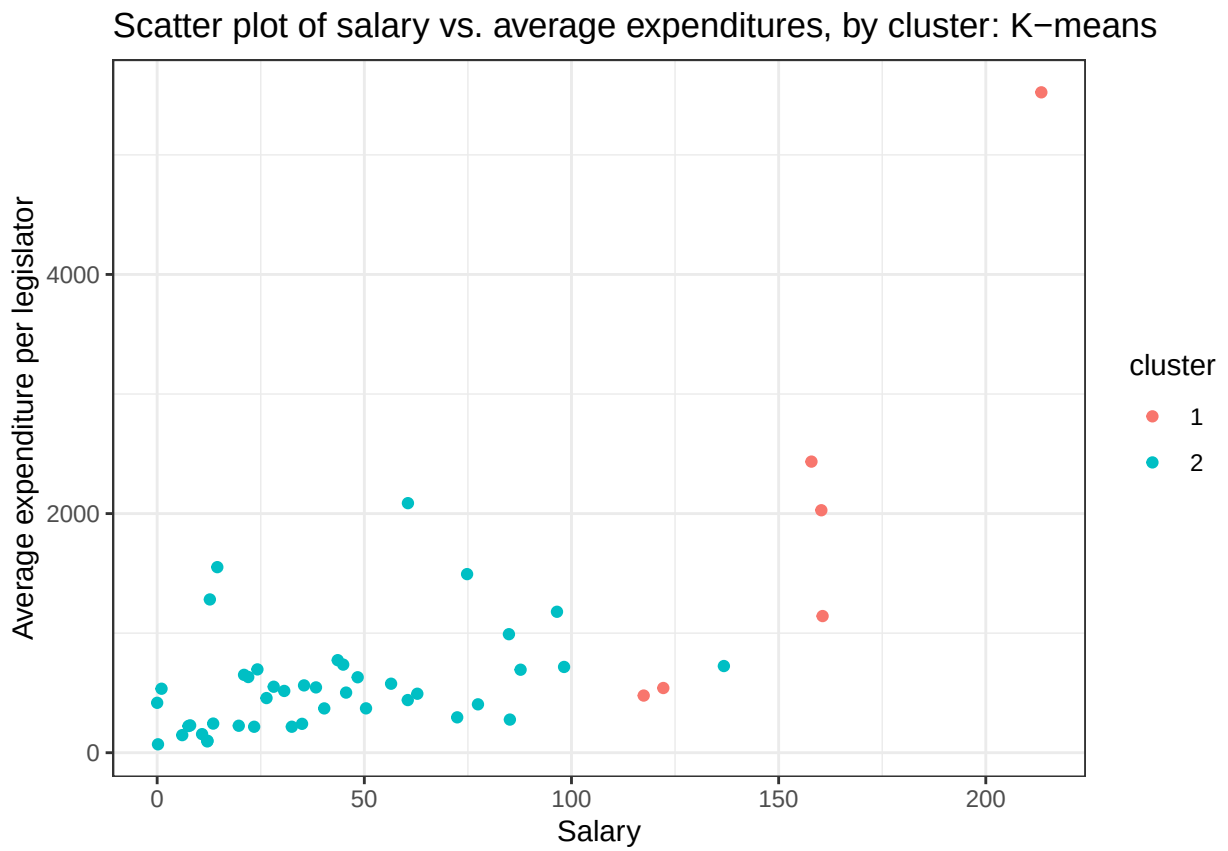


California is a very high outlier, so allowing our hierarchical clustering algorithm to make 5 clusters confines California to its own grouping. From the scatter plot we can see that dots of different colors (corresponding to states assigned to different clusters) suggests that this method with 5 clusters is not particularly insightful, although we can see distinct differences between an aggregated cluster of groups 1 and 2 and another aggregated cluster of groups 3, 4, and 5, suggesting that 2 clusters is probably the optimal number for this data.

```
# K-means
q7_plot_k <- legprof %>%
  left_join(kmeans_2_df, by = "state") %>%
  mutate(cluster = as.factor(k2)) %>%
  ggplot(aes(x = salary_real, y = expend, color = cluster)) +
  geom_point() +
  labs(x = "Salary", y = "Average expenditure per legislator") +
  theme_bw() +
  ggtitle("Scatter plot of salary vs. average expenditures, by cluster: K-means")
```

```
## Warning: Column `state` has different attributes on LHS and RHS of join
```

```
q7_plot_k
```



Question 8: Validation

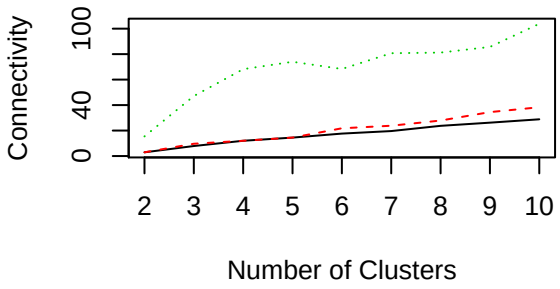
```
legprof_mat <- as.matrix(legprof[,3:6])
internal_all <- clValid(legprof_mat, 2:10,
                        clMethod = c("hierarchical", "kmeans", "model"),
                        validation = "internal")
```

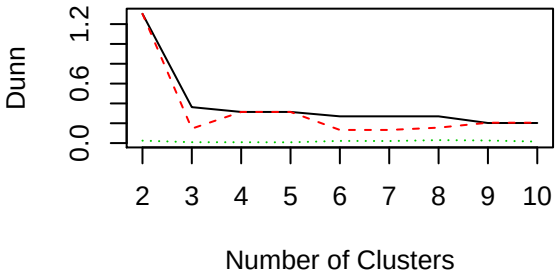
```
## Loading required package: mclust

## 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:purrr':
##
##     map
```

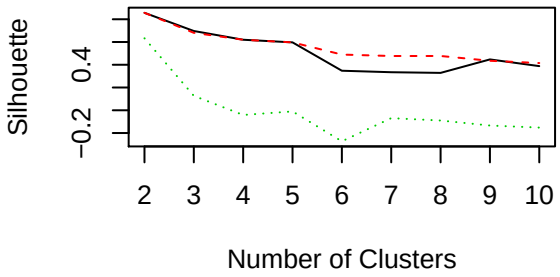
```r
summary(internal_all)
```

```
##
## Clustering Methods:
##  hierarchical kmeans model
##
## Cluster sizes:
##  2 3 4 5 6 7 8 9 10
##
## Validation Measures:
##                                  2        3        4        5        6        7        8        9
##
## hierarchical Connectivity    2.9290   7.8683  11.9841  14.4841  17.6131  19.6131  23.7075  26.1988
##              Dunn            1.3039   0.3624   0.3141   0.3141   0.2697   0.2697   0.2024
##              Silhouette      0.8567   0.6968   0.6199   0.5969   0.3475   0.3344   0.3286   0.4469
## kmeans       Connectivity    2.9290   9.6464  11.9841  14.4841  21.7869  23.7869  27.8813  34.4956
##              Dunn            1.3039   0.1459   0.3141   0.3141   0.1324   0.1324   0.1555   0.2055
##              Silhouette      0.8567   0.6809   0.6199   0.5969   0.4897   0.4769   0.4772   0.4350
## model        Connectivity   15.4179  47.0944  68.1925  73.9841  68.3488  80.7087  81.2242  85.6627
##              Dunn            0.0247   0.0087   0.0086   0.0078   0.0221   0.0207   0.0296   0.0260
##              Silhouette      0.6328   0.1271  -0.0413  -0.0101  -0.2729  -0.0692  -0.0901  -0.1342
##
## Optimal Scores:
##
##              Score  Method       Clusters
## Connectivity 2.9290 hierarchical 2
## Dunn         1.3039 hierarchical 2
## Silhouette   0.8567 hierarchical 2
```

```r
par(mfrow = c(2, 2))
plot(internal_all, legend = FALSE,
     type = "l",
     main = " ")
par(mfrow = c(1, 1))
```







In these plots, the black solid line represents hierarchical clustering, the dashed red line represents k-means, and the dotted green line represents Gaussian mixture models. On the connectivity front, a lower score indicates a good configuration of clusters, The opposite is true for the silhouette width and Dunn index, in those cases, higher values indicate good clustering. In each case for each model, it appears that 2 clusters provide the best fit.

Question 9: Discussion

Part (a)

Perhaps the biggest overarching takeaway is that it's difficult to meaningfully cluster a dataset of 49 observations. For all clustering methods and all validation metrics, 2 clusters was the optimum split, suggesting that there are few insights to be gained by arranging the data more granularly than that. This suggests that the most important distinction to be drawn in the dataset is that between the "highly professional legislatures" (CA, MA, MI, NY, OH, PA, and potentially IL depending on the method) and all other state legislatures.

Part (b)

The Gaussian mixture model performs most poorly for all cluster values on all validation metrics, so it's probably safe to say that approach is not optimal. Between the hierarchical clustering and k-means approaches, they perform roughly the same at most cluster values, and vritually identically at the optimal cluster value of 2. This optimal value of $k = 2$ holds for all configurations of models and metrics.

Part (c)

There could be a few reasons to select a sub-optimal partitioning method. Accepting only the "optimal" method precludes any insight we could gain into sub-clusters of the remaining "less professional" state legislatures outside of the "highly professional" ones. There might be insight to be gained by setting k > 2 to see how sub-clusters form, even if the validation metrics on the overall clustering are not as strong. In a more rigorous project, I would also be interested in dropping California from all analyses, as tested in Question 6, to see whether the strong cluster it formed with the other "professional" legislatures was merely a product of how much of a high outlier it was. Although the Gaussian mixture model performed the most poorly here, you might still choose to use it regardless of validation statistics in a setting where you have a strong theoretical prior that a certain variable should be composed of multiple Gaussians.