Problem Set 3

Pedro Alberto Arroyo

10/24/2019

Problem Set 3: Clustering via Partitioning

1. **Load the state legislative professionalism data from the relevant subdirectory in this repo. See the codebook for reference in the same subdirectory and combine with our discussion of these data and the concept of state legislative professionalism from class for relevant background information.**

```
load(file = "leg_pro_data/leg_pro.RData")
proDATA <- x
#head(proDATA)
```

2. **Munge the data:**

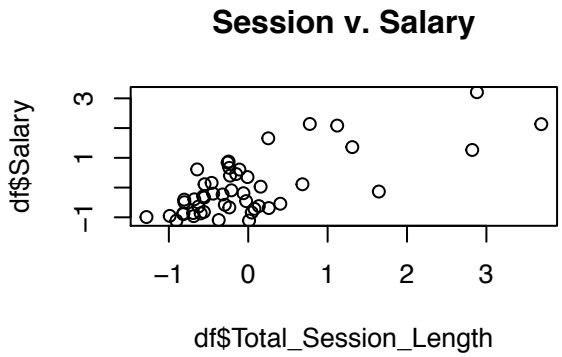
- **select only the continuous features that should capture a state legislature's level of "professionalism" (session length (total and regular), salary, and expenditures);**
- **restrict the data to only include the 2009/10 legislative session for consistency;**
- **omit all missing values;**
- **standardize the input features;**
- **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)**

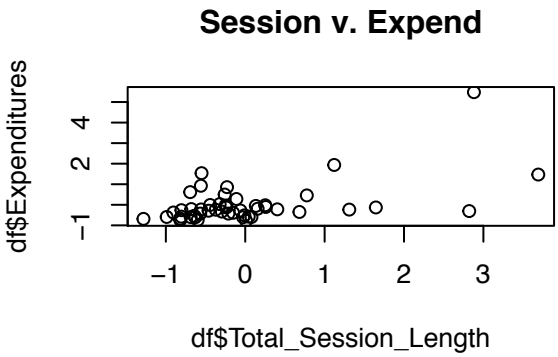
```
pro_DATA_select1 <- proDATA %>%
  dplyr::filter(sessid == "2009/10") %>%
  dplyr::select(t_slength, salary_real, expend, stateabv) %>%
  rename(Salary = salary_real) %>%
  rename(Total_Session_Length = t_slength) %>%
  rename(Expenditures = expend) %>%
  rename(State = stateabv) %>%
  drop_na()

names <- pro_DATA_select1 %>%
  dplyr::select(State)

pDM <- pro_DATA_select1 %>%
  dplyr::select(Total_Session_Length, Expenditures, Salary) %>%
  scale()

df <- data.frame(pDM)
```

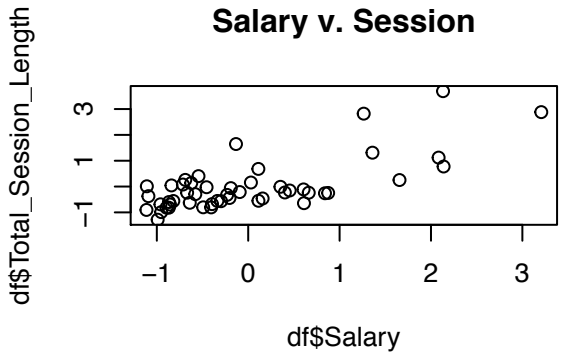
3. **Perform quick EDA visually or numerically and discuss the patterns you see.**

```
par(mfrow=c(2,2))

plot(df$Total_Session_Length, df$Salary)
title(main = "Session v. Salary")

plot(df$Total_Session_Length, df$Expenditures)
title(main = "Session v. Expend")
```

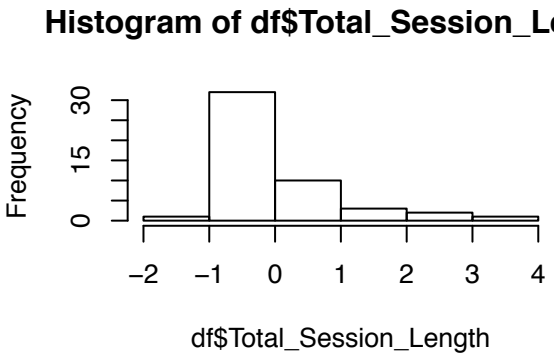
```r
plot(df$Salary, df$Total_Session_Length)
title(main = "Salary v. Session")

plot(df$Salary, df$Expenditures)
title(main = "Salary v. Expend")
```

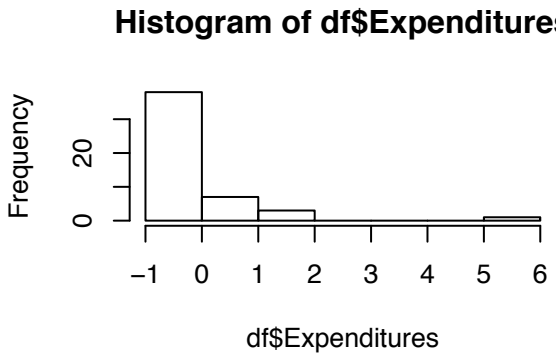


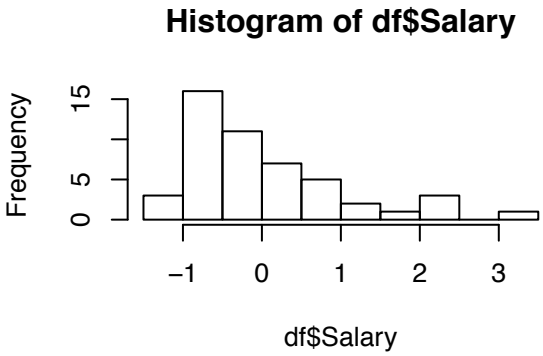





```r
par(mfrow=c(2,2))

hist(df$Total_Session_Length)
hist(df$Expenditures)
hist(df$Salary)
```



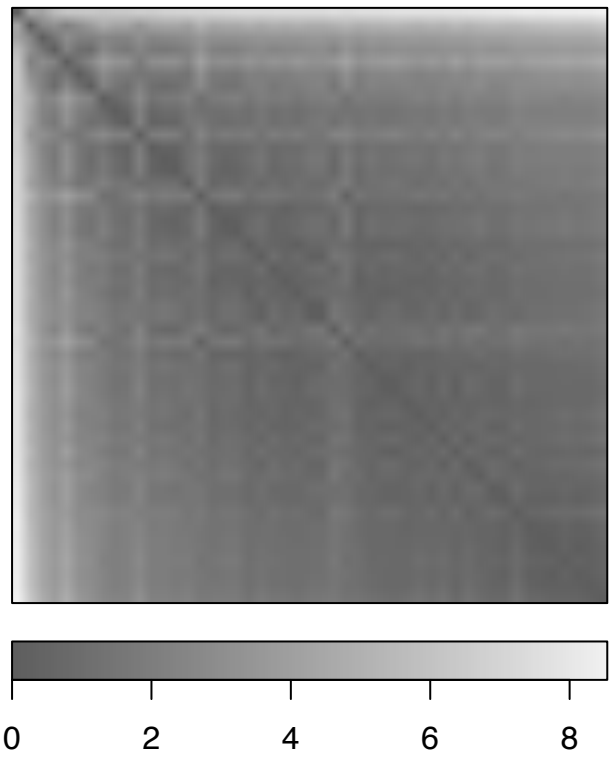




Since we're working with more than two variables, I plotted each pairwise comparison on a histogram. The three distributions seem to largely match each other, exhibiting a left-skew. This probably reflects the population skew within American states, with a lot of states that have relatively small populations and therefore operate relatively small legislatures according to the metrics we're working with. All three distributions are unimodal. Taken together, my initial reaction is that I don't expect this data to be a very good candidate for clusterability.

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.**

```
df_scaled <- scale(df)
df_dist <- dist(df_scaled)
dissplot(df_dist, main = "ODI of Pro. Leg. Data")
```

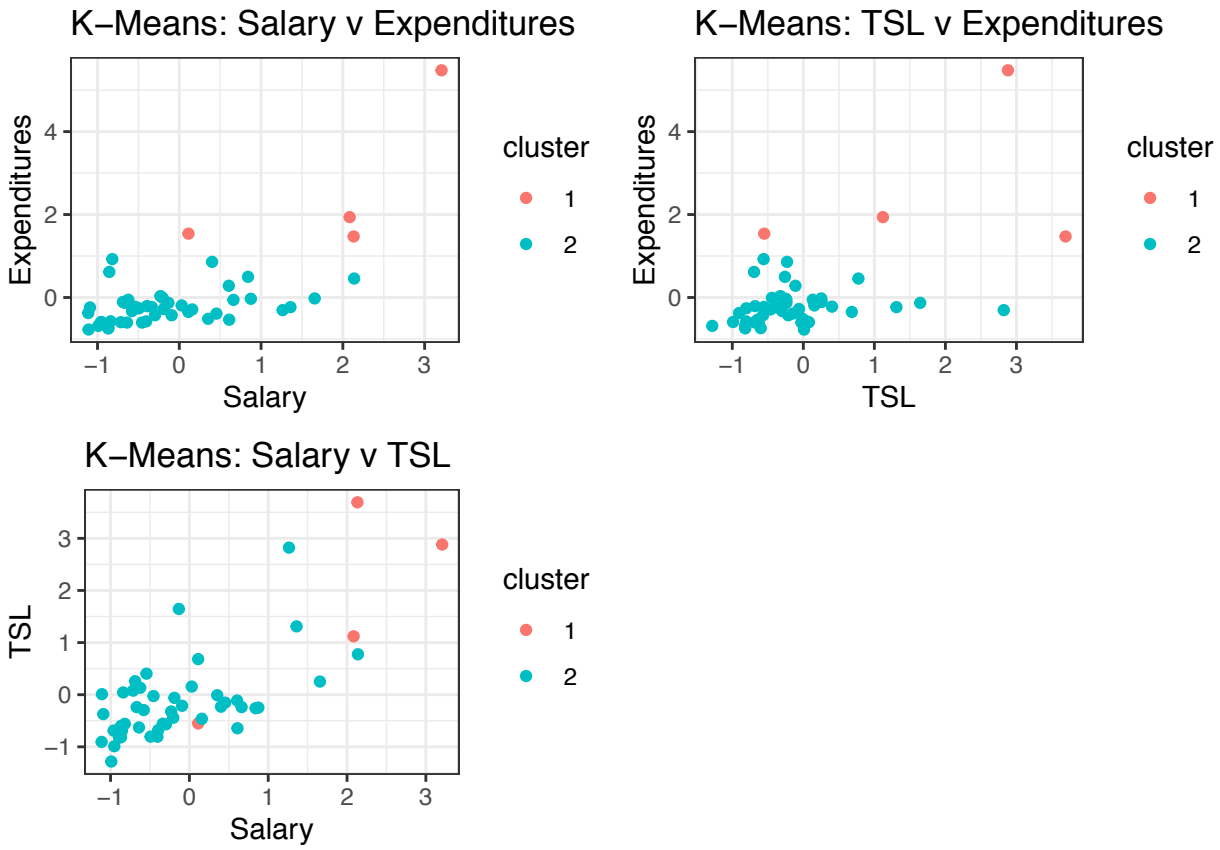


The results from the ODI analysis are not encouraging - absent pareidolia, I don't see any reason to expect that this data exhibits any significant deep structure.

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.**

```
set.seed(40713)

kmeans <- kmeans(df[,2],
                 centers = 2,
                 nstart = 15)

df$cluster <- as.factor(kmeans$cluster)
df$State <- as.factor(names$State)

k1 <- ggplot(df, aes(x=Salary, y=Expenditures, color=cluster)) +
  geom_point() +
  theme_bw() +
  scale_fill_manual(values=c("blue", "red")) +
  labs(title = "K-Means: Salary v Expenditures",
       x = "Salary",
       y = "Expenditures")

k2 <- ggplot(df, aes(x=Total_Session_Length, y=Expenditures, color=cluster)) +
  geom_point() +
  theme_bw() +
  scale_fill_manual(values=c("blue", "red")) +
```

```
  labs(title = "K-Means: TSL v Expenditures",
       x = "TSL",
       y = "Expenditures")

k3 <- ggplot(df, aes(x=Salary, y=Total_Session_Length, color=cluster)) +
  geom_point() +
  theme_bw() +
  scale_fill_manual(values=c("blue", "red")) +
  labs(title = "K-Means: Salary v TSL",
       x = "Salary",
       y = "TSL")

plot_grid(k1, k2, k3)
```




```
which(df$Expenditures > 5)
```

```
## [1] 5
```

```
df[5, ]
```

```
##   Total_Session_Length Expenditures   Salary cluster State
## 5             2.880726     5.478545 3.206991       1    CA
```

```
df2 <- df[-c(5),]

kmeans$size
```

```
## [1]  4 45
```

The k-means algorithm produced two clusters, on of which contains many more members than the other (45:4). I plotted those clusters against three scatterplots, representing the three pairwise comparisons contained in the data. There seems to be a loose pattern, with cluster 2 representing low values among all dimensions and cluster 1 (generally) representing high values. The Salary v. Expenditures plot shows a fairly dramatic outlier. I did not remove it for two reasons: (i) it is less dramatic of an outlier in the other two plots, (ii) it's California, and any effort to model American states would feel incomplete without the golden state.

Broadly speaking, though, there are arguably a set of outliers that might be removed. If I were working with a dataset that seemed to be a better candidate for clusterability, or if I was working with a large enough dataset to feel confident that slight structures might represent underlying dynamics, I might pursue that as a strategy. But, as it stands, with one year's data for 50 cases and low apparent clusterability, it would start to feel like I'm forcing the data into a pattern that isn't meaningful.

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.**

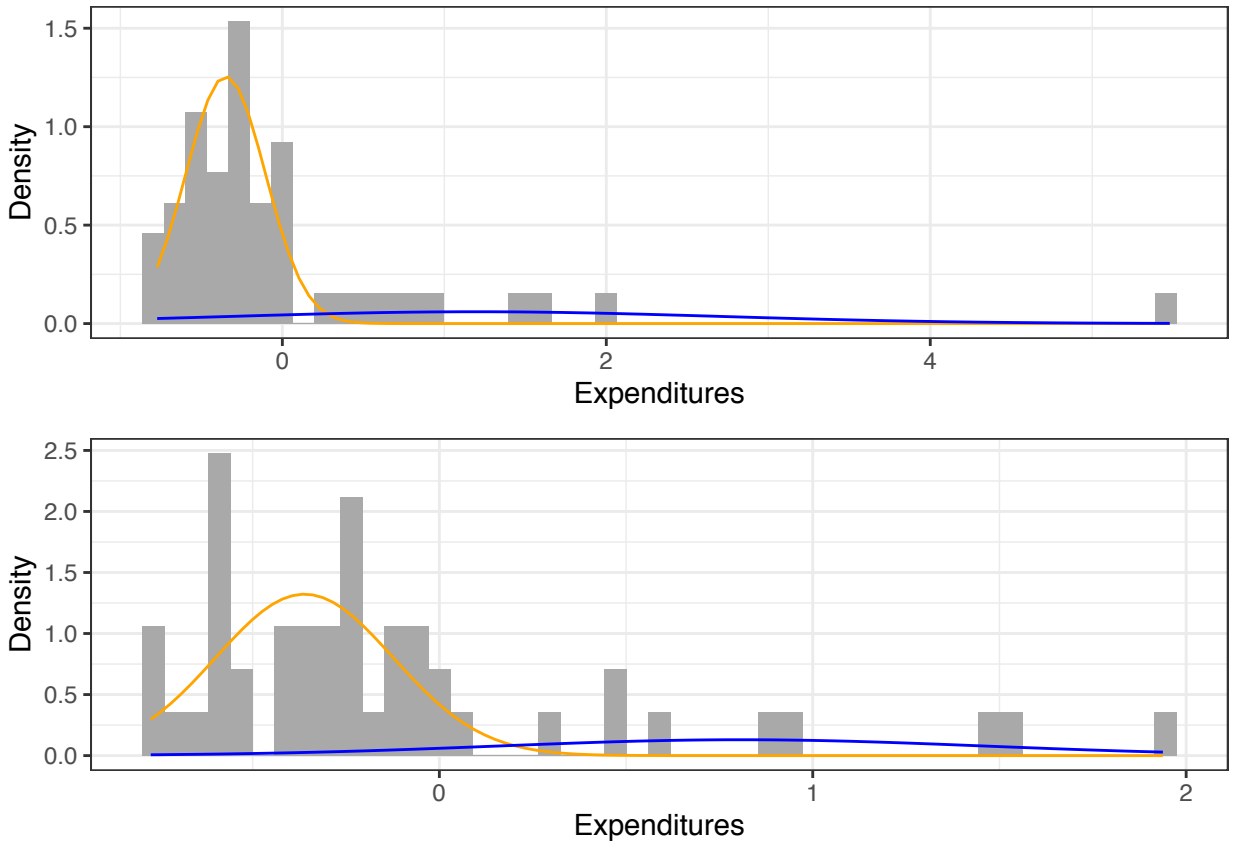
```
par(mfrow=c(1,1))

set.seed(7200)
gmm <- normalmixEM(df$Expenditures,
                   k = 2)
```

```
## number of iterations= 17
```

```
g1 <- ggplot(data.frame(x = gmm$x)) +
  geom_histogram(aes(x, ..density..), fill = "darkgray", bins=48) +
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm$mu[1], gmm$sigma[1], lam = gmm$lambda[1]),
                colour = "orange") +
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm$mu[2], gmm$sigma[2], lam = gmm$lambda[2]),
                colour = "blue") +
  xlab("Expenditures") +
  ylab("Density") +
  theme_bw()

posterior <- data.frame(cbind(gmm$x, gmm$posterior))
posterior$component <- ifelse(posterior$comp.1 > 0.3, 1, 2)
ggg <- df
ggg$cluster3 <- as.factor(posterior$component)

ggg1 <- ggplot(ggg, aes(x=Salary, y=Total_Session_Length, color=cluster3)) +
  geom_point() +
  theme_bw() +
  scale_fill_manual(values=c("red", "blue")) +
  labs(title = "Gaussian Mixture Model: Salary v TSL",
       x = "Salary",
       y = "TSL")

set.seed(7200)

gmm2 <- normalmixEM(df2$Expenditures,
                    k = 2)
```

```
## number of iterations= 29
```

```
g2 <- ggplot(data.frame(x = gmm2$x)) +
  geom_histogram(aes(x, ..density..), fill = "darkgray", bins=47) +
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm2$mu[1], gmm2$sigma[1], lam = gmm2$lambda[1]),
                colour = "orange") +
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm2$mu[2], gmm2$sigma[2], lam = gmm2$lambda[2]),
                colour = "blue") +
  xlab("Expenditures") +
  ylab("Density") +
  theme_bw()

plot_grid(g1, g2, ncol = 1)
```

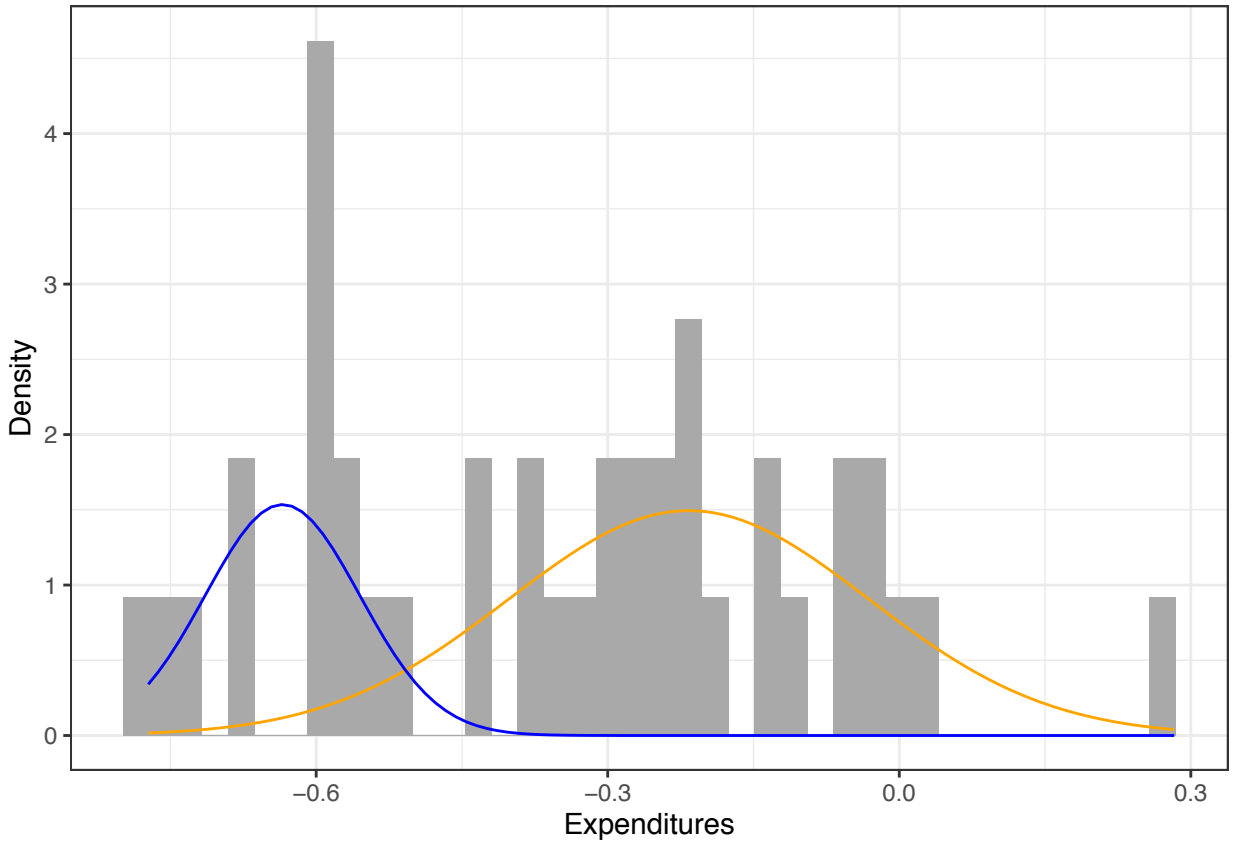


The GMM tells a very similar story. One of the two clusters (in yellow) seems to be better-defined than the other. In this representation, the other cluster seems to mostly be trying to capture the residual. One of the items represented a dramatic outlier, and so I removed it. The second of the two plots reports the result. This model begins to suggest a two-clusters-plus-outliers structure. So, I removed the cases from the right side of the distribution and plotted the new GMM:

```
set.seed(7200)

df3 <- df2[-c(2,8,21,27,29,31,37,42),]

gmm3 <- normalmixEM(df3$Expenditures,
                    k = 2)
```

```
## number of iterations= 138
```

```
ggplot(data.frame(x = gmm3$x)) +
  geom_histogram(aes(x, ..density..), fill = "darkgray", bins=40) +
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm3$mu[1], gmm3$sigma[1], lam = gmm3$lambda[1]),
                colour = "orange") +
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm3$mu[2], gmm3$sigma[2], lam = gmm3$lambda[2]),
                colour = "blue") +
  xlab("Expenditures") +
  ylab("Density") +
  theme_bw()
```



As expected, this starts to show a little more 'clustering'. But, in order to get even this modest result, we had to drop: AK, FL, MI, NV, NJ, NY, PA, TX, & CA. The number of iterations has also increased throughout, from 17 to 29 to 138.

7. **Fit one additional partitioning technique of your choice (e.g., PAM, CLARA, fuzzy C-means, DBSCAN, etc.), and present and discuss results. Here again initialize at k=2.**

```
Pp <- pam(df, 2, diss = FALSE,
    metric = c("euclidean", "manhattan"),
    medoids = NULL, stand = FALSE, cluster.only = TRUE,
    do.swap = TRUE,
    trace.lev = 0)

Pp <- data.frame(Pp)
ppp <- df
ppp$cluster2 <- as.factor(Pp$Pp)
```

```r
ggplot(ppp, aes(x=Salary, y=Total_Session_Length, color=cluster2)) +
  geom_point() +
  theme_bw() +
  scale_fill_manual(values=c("blue", "red")) +
  labs(title = "PAM Partition: Salary v TSL",
       x = "Salary",
       y = "TSL")
```


```r
ppp1 <- ggplot(ppp, aes(x=Salary, y=Total_Session_Length, color=cluster2)) +
  geom_point() +
  theme_bw() +
  scale_fill_manual(values=c("blue", "red")) +
  labs(title = "PAM Partition: Salary v TSL",
       x = "Salary",
       y = "TSL")
```

I chose the PAM approach for this exercise. Though it is similar to the k-means approach, the results look pretty different. The most interesting thing is that the clusters seem to be intertwined - so to the extent that this is picking out different clusters, it's doing it in a way that doesn't map on to the way these data points are represented when reduced to two dimensions. (Though not shown, I did run this with different variables in the x and y position, and the pattern persisted.)

I've already said that this data doesn't look very clusterable to me, so I should probably leave that there. But, I have to admit that the non-intuitive presentation of these clusters piques my curiosity a bit.

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).**

```
plot_grid(k3, ppp1, ggg1, ncol = 1)
```

It's interesting to compare these side by side. It looks like the K-Means approach produces the most pleasing clusters, in that they are the clusters that appear to be the most distinct. The presence of one item 'out of place', as it were, in the kmeans model is a product of the axes of the plot: from the earlier example, we can see that this data point is not intertwined when different axes are used. All in all, the k-means algorithm with not outliers removed and with two clusters seems to do a fairly good job of capturing the minimal structure that we so far have reason to believe is there; namely, that a few of the states are acting in a wierd way when compared to the others.

To satisfy my curiosity, I used the which() function to get the names of those states. They are CA, FL, NY, PA. It seems interesting that those are fairly large states. In fact, they're 4 out of the 5 biggest states. That sort of makes the absence of the fifth state in that cluster pretty interesting - specially because that state is Texas.

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).**

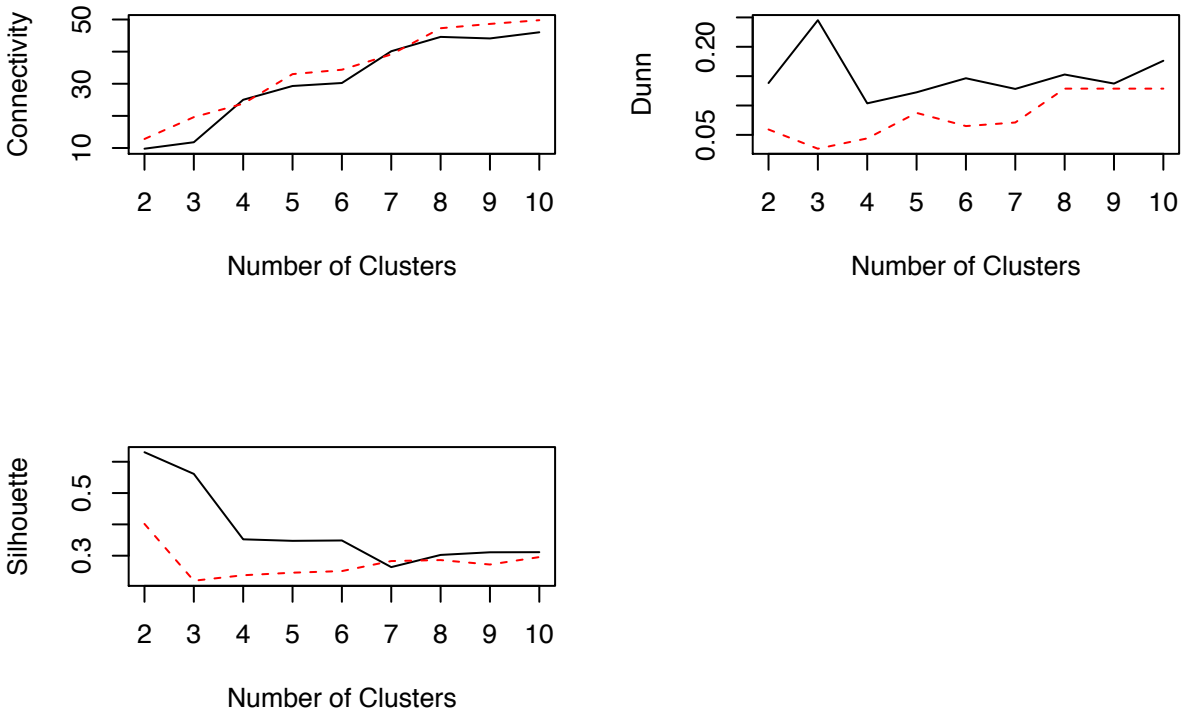
I used the clValid package to check validity in the PAM and KMEANS models, focusing on the silhouette measure. The results seem to indicate that the optimal number of clusters is likely 2, with some support for 3. I was not able to check validity for the GMM model.)

```
int <- dplyr::select(df, Total_Session_Length, Salary, Expenditures)
check <- clValid(int, 2:10,
                 clMethods = c("kmeans", "pam"),
                 validation = "internal"); summary(check)
```

```
##
```

```
## 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   9.7798 11.7972 25.0143 29.3016 30.2357 40.1242 44.5897 44.1071 46.0250
##        Dunn           0.1383  0.2453  0.1037  0.1225  0.1464  0.1282  0.1528  0.1373  0.1762
##        Silhouette     0.6304  0.5612  0.3522  0.3474  0.3486  0.2635  0.3024  0.3109  0.3112
## pam    Connectivity  12.8083 19.6091 23.7881 33.0008 34.3671 39.1282 47.2877 48.6210 49.7857
##        Dunn           0.0592  0.0264  0.0440  0.0874  0.0650  0.0710  0.1287  0.1287  0.1287
##        Silhouette     0.4015  0.2200  0.2376  0.2457  0.2508  0.2829  0.2860  0.2718  0.2955
##
## Optimal Scores:
##
##              Score  Method Clusters
## Connectivity 9.7798 kmeans 2
## Dunn         0.2453 kmeans 3
## Silhouette   0.6304 kmeans 2
```

```r
par(mfrow = c(2, 2))

plot(check, legend = FALSE,
     type = "l",
     main = " ")
```



10. **Discuss the validation output.**

- **What can you take away from the fit?**
- **Which approach is optimal? And optimal at what value of k?**
- **What are reasons you could imagine selecting a technically "sub-optimal" partitioning method, regardless of the validation statistics?**

The validation results support the contention that the best clustering model to use for this dataset is a kmeans model at k=2. That seems to be the most parsimonious representation of the minimal underlying structure that seems to be there.

Still, I decided to follow up on the possibility of three clusters:

```
set.seed(40713)

kmeans <- kmeans(df[,3],
                 centers = 3,
                 nstart = 15)

df$cluster3 <- as.factor(kmeans$cluster)
df$State <- as.factor(names$State)

k4 <- ggplot(df, aes(x=Salary, y=Expenditures, color=cluster3)) +
  geom_point() +
  theme_bw() +
  scale_fill_manual(values=c("blue", "red", "green")) +
  labs(title = "K-Means: Salary v Expenditures",
       x = "Salary",
       y = "Expenditures")

plot_grid(k1, k4)
```

I was genuinely surprised to see that the three-cluster approach produced such an interesting distribution. At this point, I would have to reconsider some of my earlier skepticism about whether or not this data is, in fact, well-described by a cluster model. There seem to be three d istinct *columns*, within which the same general trend repeats. At this point, if this were an ongoing project, I would want to step back and take a fresh look at things to try get a sense of whether I think that this new model is revealing something meaningful. Again, I think that the only way to engage with that question in a real-world setting would be within the context of a larger research project.

On the question of when it might be valuable to select a sub-optimal model, I think that it would likely come down to domain knowledge or data validity concerns. In a lot of 'real-world' applications, it's going to be the case that there are a limited number of 'natural kinds' that you're interested in as a researcher. If you're tyring to build an explanatory model, it might very well be the case that a simpler and more parsimonious model is prefered over an unwieldly alternative, even if the latter is technically superior.