Intro to ML Hwk 4

William Ansehl

3/2/2020

R Markdown

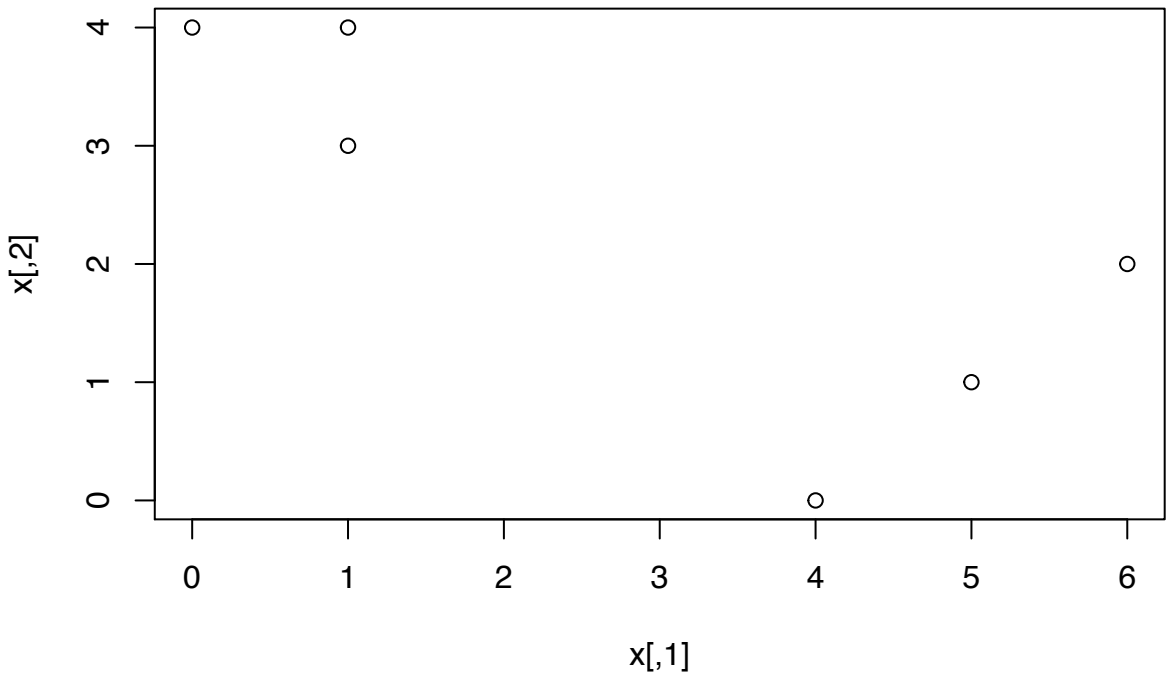
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Performing K-Means by Hand

Problem 1

```
x <- cbind(c(1, 1, 0, 5, 6, 4), c(4, 3, 4, 1, 2, 0))
plot(x)
```

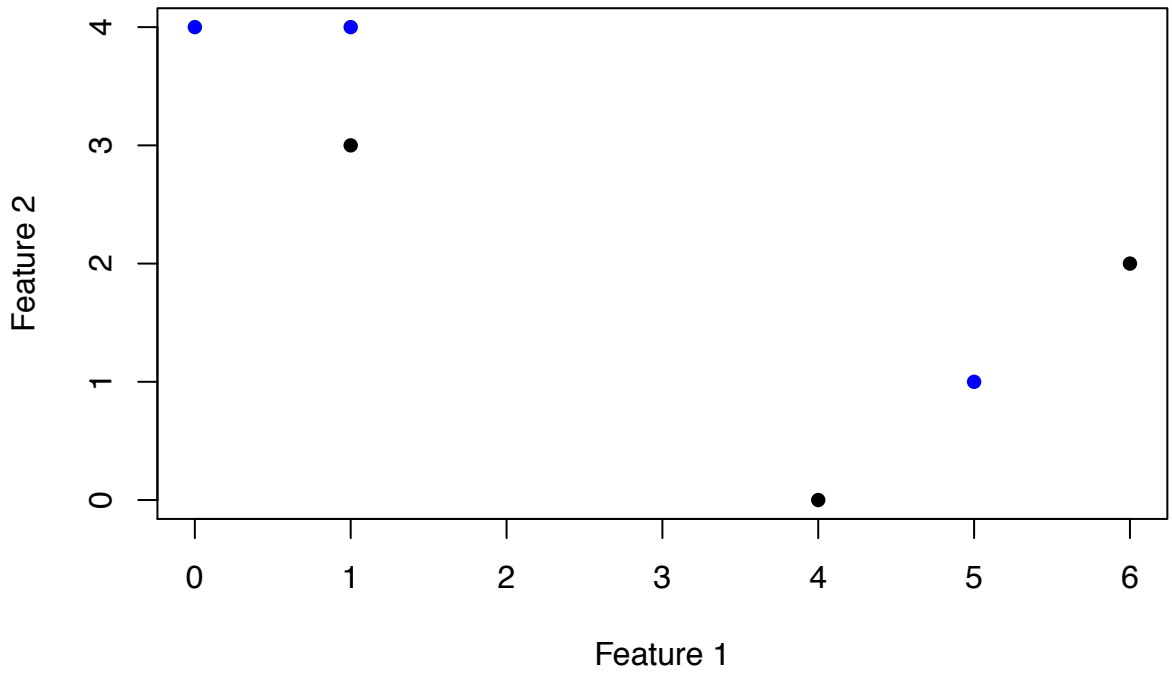


Problem 2

```r
set.seed(100)
(cluster.label <- sample(seq(0,1), size = nrow(x), replace = TRUE))
```

```
## [1] 1 0 1 1 0 0
```

```r
plot(x[,1],x[,2],col=rgb(0,0,cluster.label),pch=16,xlab="Feature 1",ylab="Feature 2")
```



Problem 3

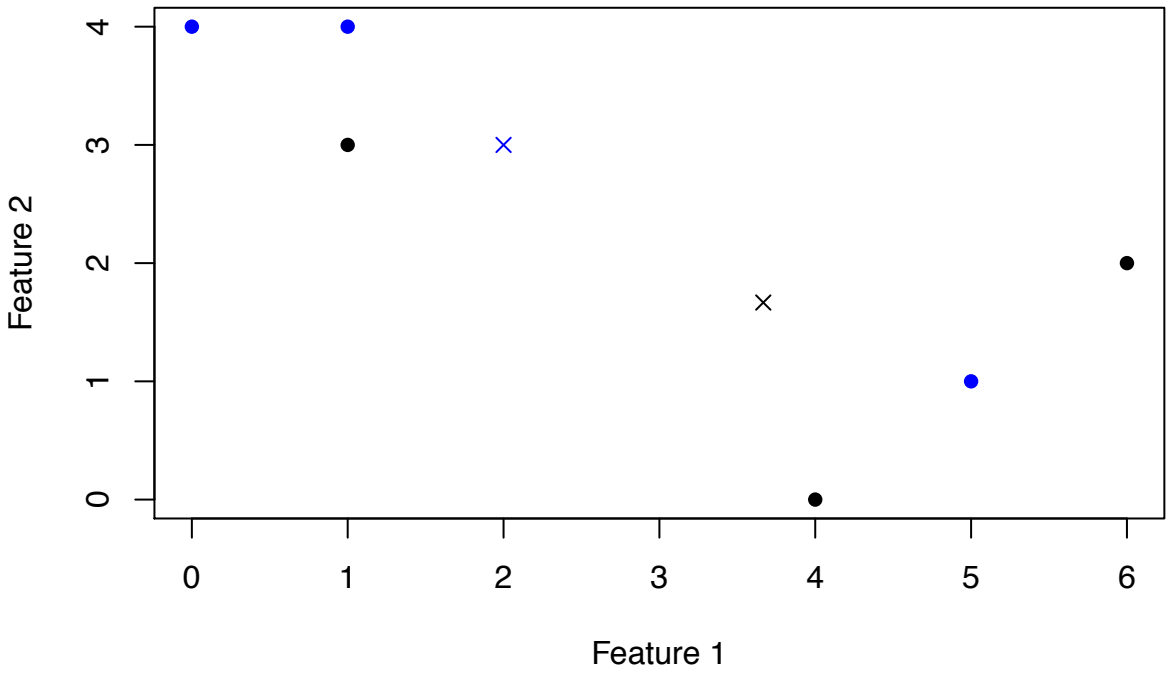
```r
x0 = mean(x[cluster.label == 0, 1])
y0 = mean(x[cluster.label == 0, 2])
x1 = mean(x[cluster.label == 1, 1])
y1 = mean(x[cluster.label == 1, 2])
(centroid0 = c(x0,y0))
```

```
## [1] 3.666667 1.666667
```

```r
(centroid1 = c(x1,y1))
```

```
## [1] 2 3
```

```r
# Centroid for cluster 0 is [3.67,1.67]
# Centroid for cluster 1 is [2, 3]
plot(x[,1],x[,2],col=rgb(0,0,cluster.label),pch=16,xlab="Feature 1",ylab="Feature 2")
points(centroid0[1], centroid0[2], col = "black", pch = 4)
points(centroid1[1], centroid1[2], col = "blue", pch = 4)
```




```
# The "x"'s mark the centroids of their respective clusters
```

Problem 4/5

```
distance <- function (x, y){
  return(sqrt((x[1] - y[1])^2 + (x[2] - y[2])^2))
}

close_clust <- NULL
table <- cbind(x, cluster.label)
for(i in 1:nrow(x)){
  close_clust[i] <- if(distance(x[i,], centroid0) <= distance(x[i,], centroid1)) 0 else 1
}
(table <- cbind(table, close_clust))
```

```
##          cluster.label close_clust
## [1,] 1 4             1           1
## [2,] 1 3             0           1
## [3,] 0 4             1           1
## [4,] 5 1             1           0
## [5,] 6 2             0           0
## [6,] 4 0             0           0
```

```
# We see that the second point was closer to cluster 1 than cluster 0.
# We alse see that the fourth point was closer to cluster 0 than cluster 1.

x0 <-mean(table[,1][table[,4] == 0])
y0 <-mean(table[,2][table[,4] == 0])
x1 <-mean(table[,1][table[,4] == 1])
y1 <-mean(table[,2][table[,4] == 1])
(centroid0 = c(x0,y0))
```

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

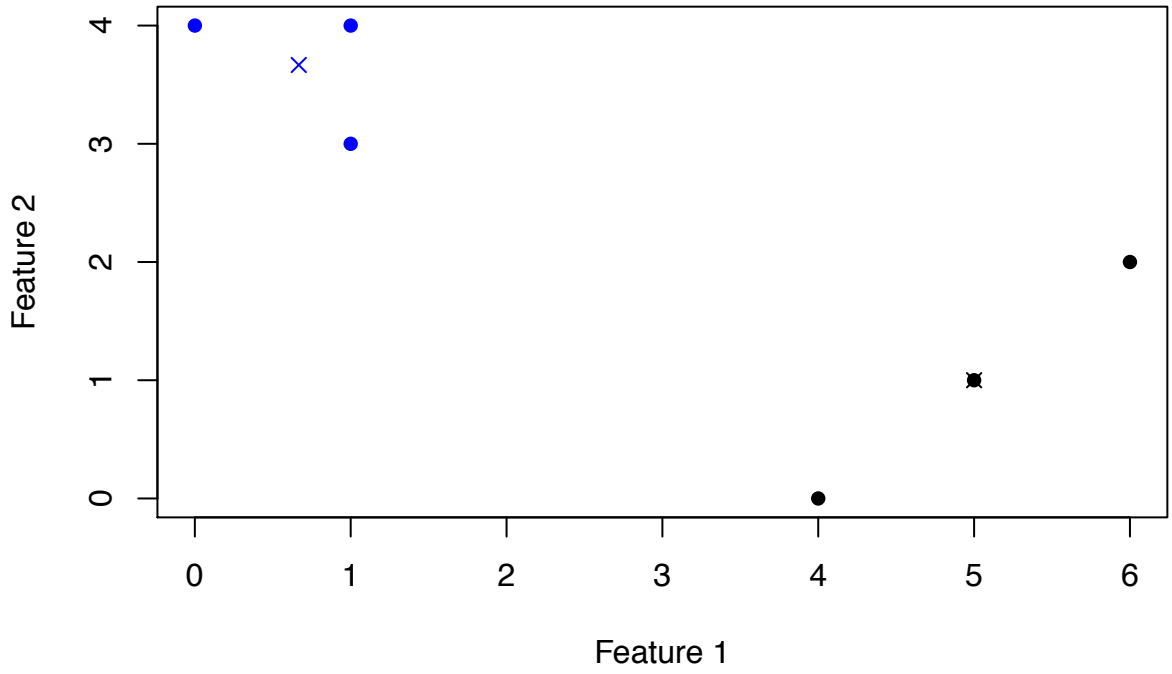
```
(centroid1 = c(x1,y1))
```

```
## [1] 0.6666667 3.6666667
```

```
# Nothing would change if we proceeded through an iteration, the labels
# to the clusters would not change.
```

Problem 6

```
plot(x[,1],x[,2],col=rgb(0,0,close_clust),pch=16,xlab="Feature 1",ylab="Feature 2")
points(centroid0[1], centroid0[2], col = "black", pch = 4)
points(centroid1[1], centroid1[2], col = "blue", pch = 4)
```




```
# The "x"'s on the graph represent the centroids for their respective clusters
```

Clustering State Legislative Professionalism

Problem 1

```
load("legprof-components.v1.0.RData" )
data <- x
```

Problem 2

```
data_reduced <- na.omit(data[data$year == '2009' | data$year == '2010',]) # (b)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
df <- data_reduced %>% # (a), (c) and (d)
  select(t_slength, slength, salary_real, expend) %>%
  drop_na() %>%
  scale()

(rownames(df) <- data_reduced$state) #e, associated state names
```

```
##  [1] "Alabama"        "Alaska"         "Arizona"        "Arkansas"
##  [5] "California"     "Colorado"       "Connecticut"    "Delaware"
##  [9] "Florida"        "Georgia"        "Hawaii"         "Idaho"
## [13] "Illinois"       "Indiana"        "Iowa"           "Kansas"
## [17] "Kentucky"       "Louisiana"      "Maine"          "Maryland"
## [21] "Massachusetts"  "Michigan"       "Minnesota"      "Mississippi"
## [25] "Missouri"       "Montana"        "Nebraska"       "Nevada"
## [29] "New Hampshire"  "New Mexico"     "New York"       "North Carolina"
## [33] "North Dakota"   "Ohio"           "Oklahoma"       "Oregon"
## [37] "Pennsylvania"   "Rhode Island"   "South Carolina" "South Dakota"
## [41] "Tennessee"      "Texas"          "Utah"           "Vermont"
## [45] "Virginia"       "Washington"     "West Virginia"  "Wyoming"
```

Problem 3

```
library(factoextra)
```

```
## Loading required package: ggplot2
```

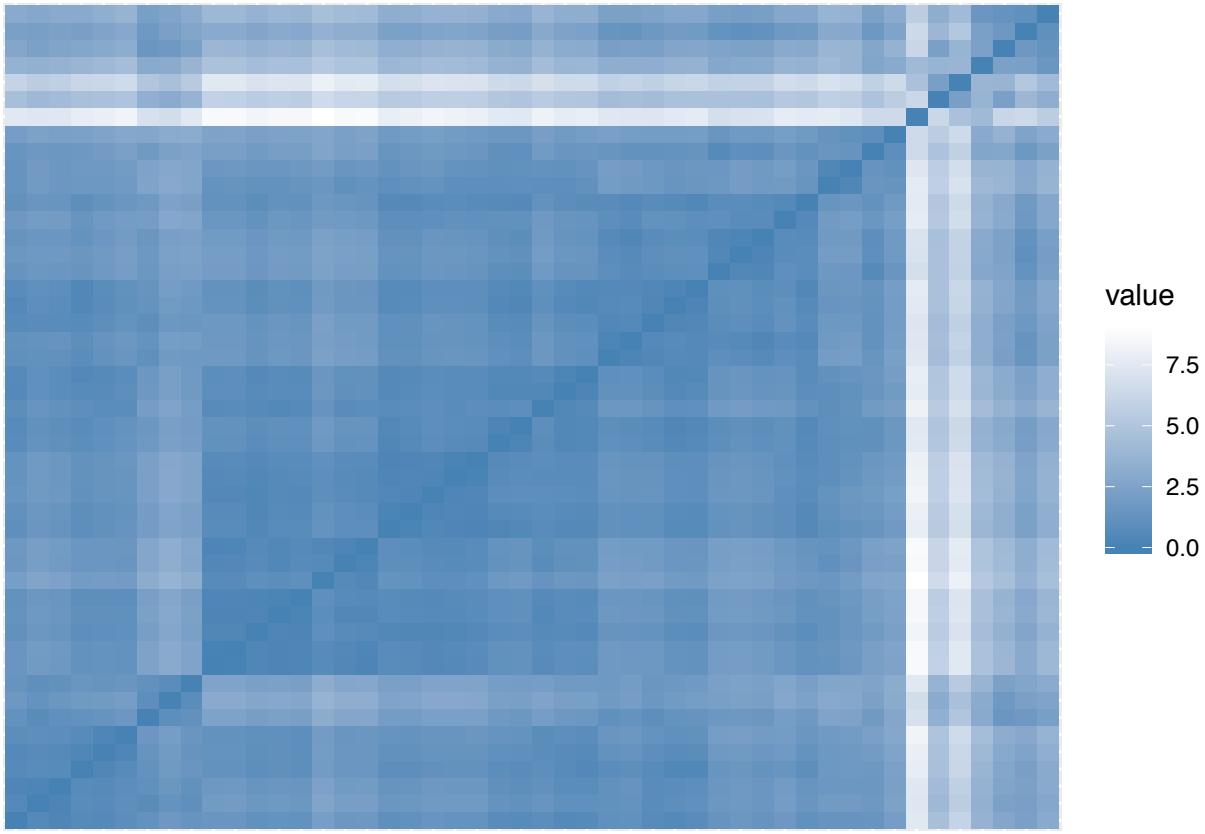
```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
clust <- get_clust_tendency(df, 30)
clust$hopkins_stat
```

```
## [1] 0.7831716
```

```
clust$plot +
  scale_fill_gradient(low = "steelblue", high = "white")
```

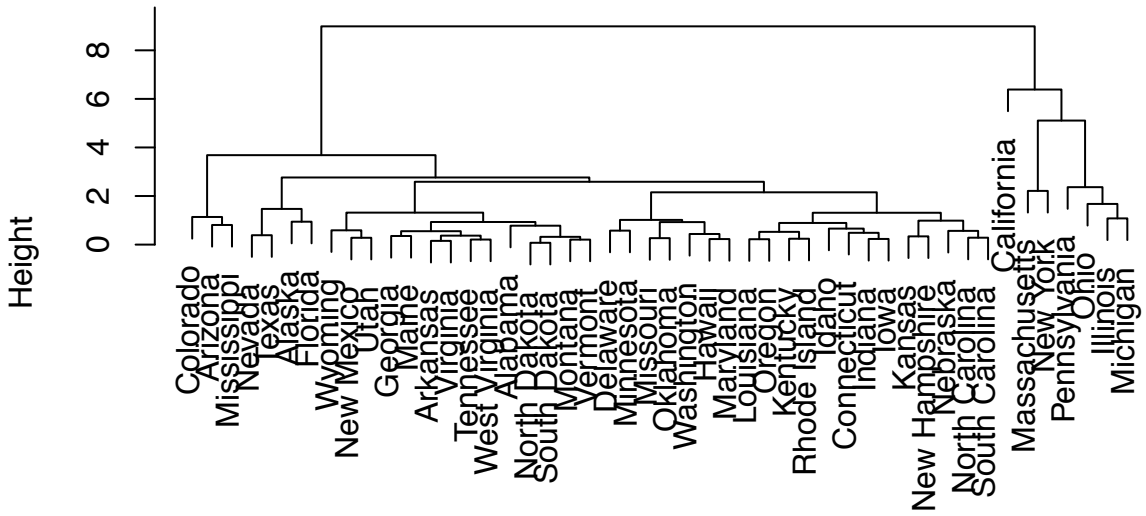
```
## Scale for 'fill' is already present. Adding another scale for 'fill', which
## will replace the existing scale.
```




```
# The hopkins value is approx. 0.78, which indicates that the data is
# relatively highly clusterable. Random data will have a hopkins value
# of 0.5 and uniform data will have a haopkins value close to 0.
# A hopkins value close to 1 indicates that the data is highly clustered.

# In the clusterability plot, blue indicates low dissimilarity values and
# white indicates high dissimilarity values. The color scheme is proportional
# to the values of dissimilarity among observations.
# Since we see a lot of blue (darker blue esspecially) within the plot,
# the plot therefore suggests high levels of similarity between observations
# (and thus clusterability).
```

Problem 4

```
set.seed(101)
hc_complete <- hclust(stats::dist(df), method = "complete")
plot(hc_complete)
```



stats::dist(df)
hclust (*, "complete")

```
# Hierarchicaly clustered the data with complete linkage methodology.
# california, Massachusetts, New York, Pennsylvania and Colorado are
# highly dissimilar from other clusters.
# We still see typical relationships we would expect,
# such as North and South Carolina clustered together
# as well as Noth and South Dakota.
# We also see a few expected regional similarities such as in the midwest with
# Missouri and Oaklahoma.
# we also see a number of currently inexplainable relationships
# such as Hawaii being clustered with Maryland and Washington
```

Problem 5

```
kmeans <- kmeans(df,
                 centers = 2,
                 nstart = 15)
output <- data.frame(as.table(kmeans$cluster))
freq2 <- output[output$Freq == "1",]
# California, Massachusetts, Michigan, New York, Ohio and Pennsylvania
# are classified as cluster 1 while the rest of the states are cluster 2.
kmeans$centers
```

```
##    t_slength    slength salary_real     expend
## 1  2.0079549  2.0643454     2.04323  1.4647791
## 2 -0.2868507 -0.2949065    -0.29189 -0.2092542
```

```
# We also see that cluster 1 has higher levels of continuous variables
# such at every level (salary, expenditures, session length).
```

```
# Hence, cluster has higher levels of professionalism.
```

Problem 6

```
set.seed(7355)
library(mixtools)
```

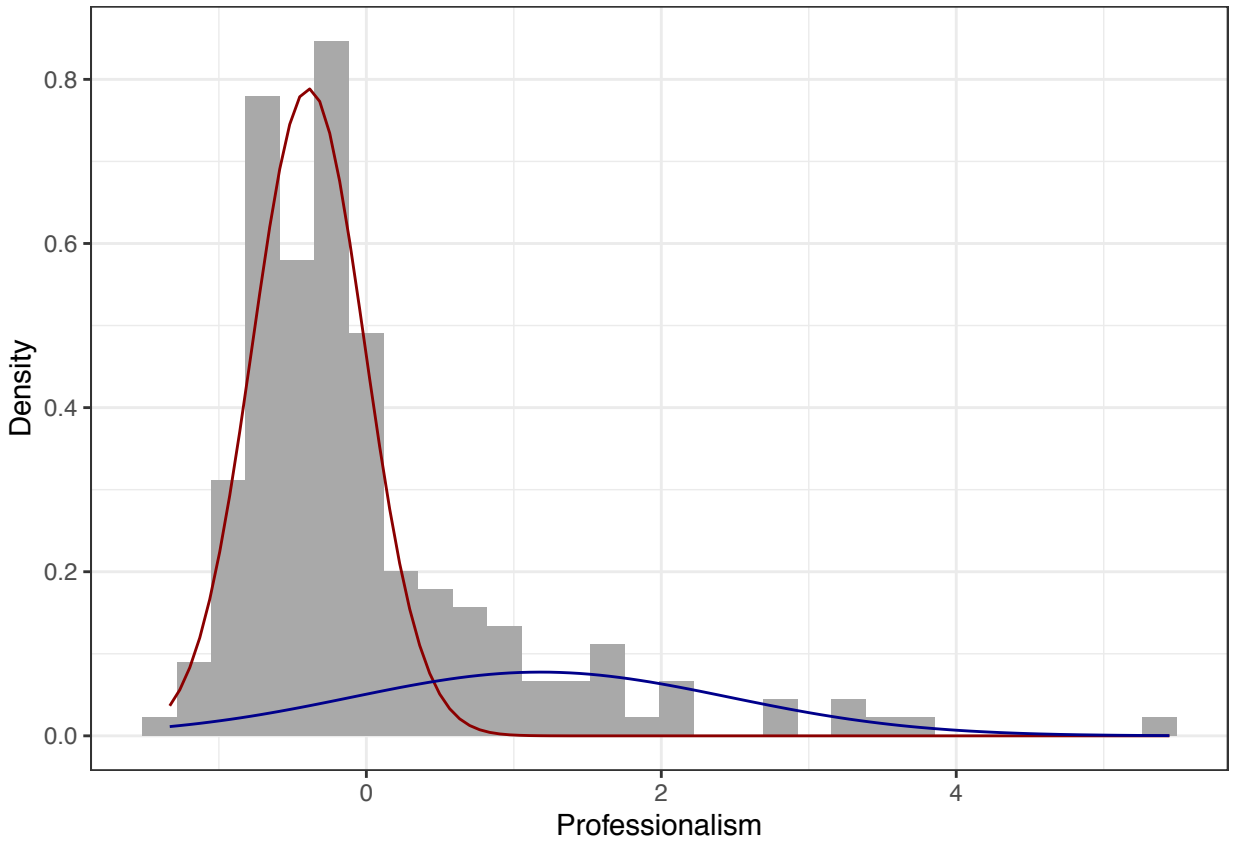
```
## mixtools package, version 1.2.0, Released 2020-02-05
## This package is based upon work supported by the National Science Foundation under Grant No. SES-0518
```

```
gmm1 <- normalmixEM(df, k = 2)
```

```
## number of iterations= 38
```

```
library(plotGMM)
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 = "darkred") +
  stat_function(geom = "line", fun = plot_mix_comps,
                args = list(gmm1$mu[2], gmm1$sigma[2], lam = gmm1$lambda[2]),
                colour = "darkblue") +
  xlab("Professionalism") +
  ylab("Density") +
  theme_bw()
```

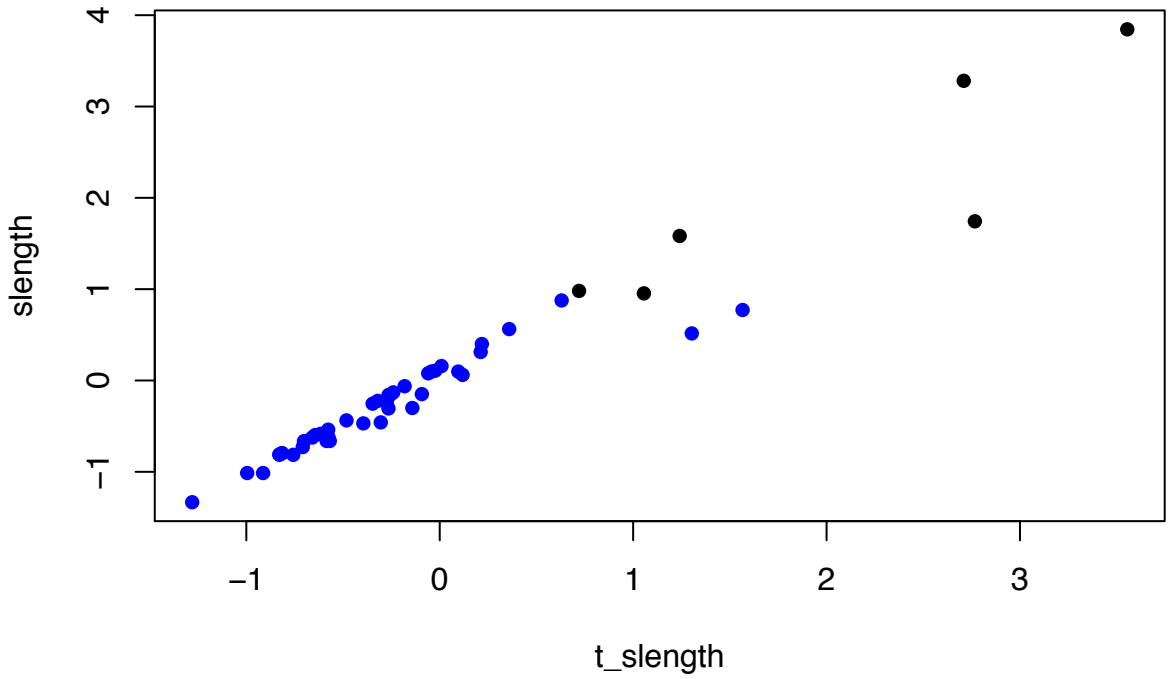
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



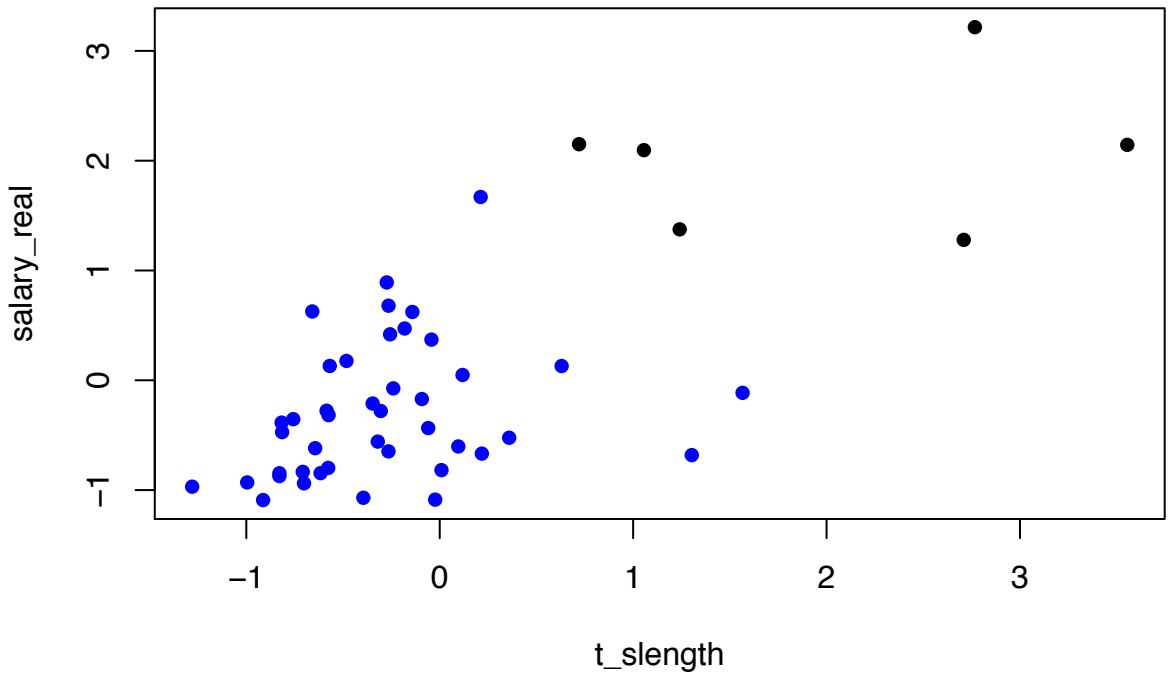

```
# Cluster 1 (indicated by the red line) has a centroid a little smaller than
# 0 and is highly clustered together.
# Cluster 2 (blue line) has a centroid greater than 0 and is much more spread
# apart, with overall higher levels of professionalism
# than cluster 1.
```

Problem 7

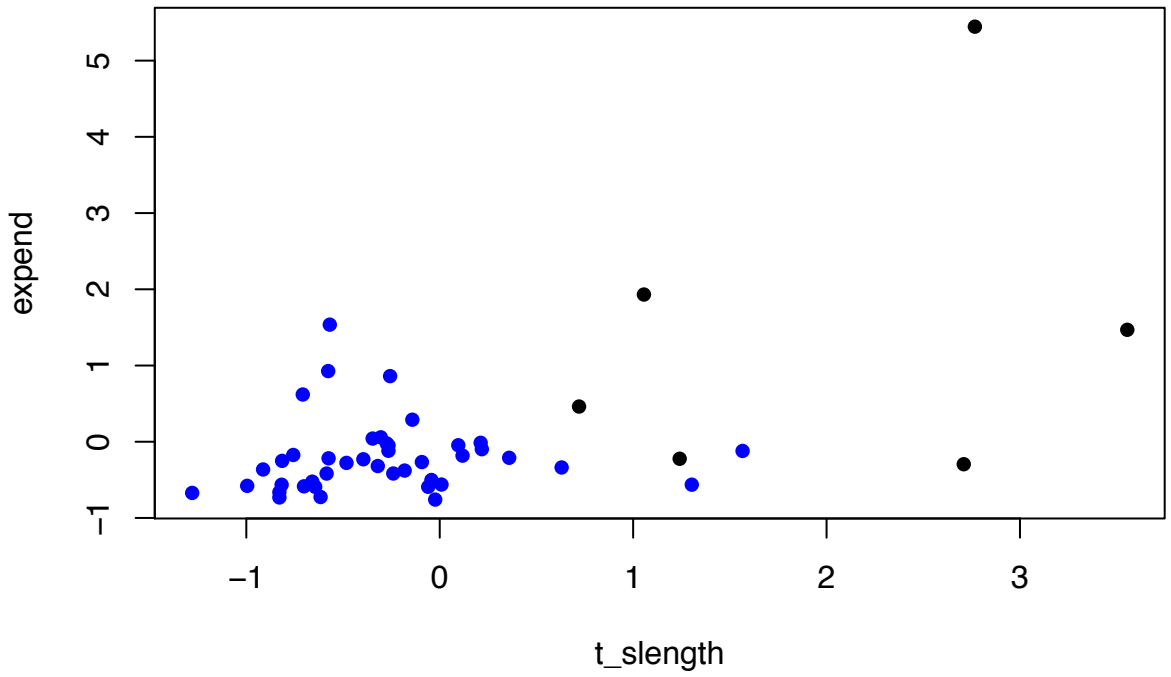
```
rownames(output) <- data_reduced$state
output$Var1 <- NULL
colnames(output)[colnames(output)=="Freq"]<-"Cluster"

merged <- merge(df,output,by.x = 0, by.y = 0)
merged["Cluster"] = merged["Cluster"]-1

# t_slength by slength
plot(merged[,2],merged[,3],col=rgb(0,0,merged[,"Cluster"]),pch=16,xlab="t_slength",ylab="slength")
```



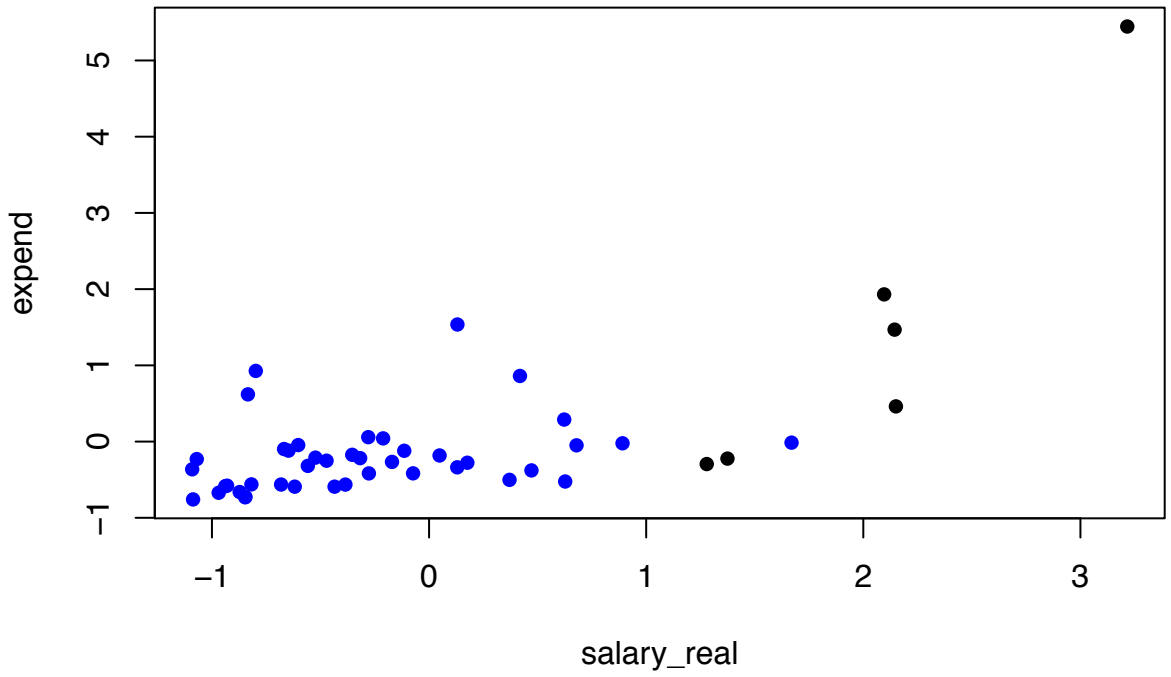

```
# t_slength by salary
plot(merged[,2],merged[,4],col=rgb(0,0,merged[,"Cluster"]),pch=16,xlab="t_slength",ylab="salary_real")
```



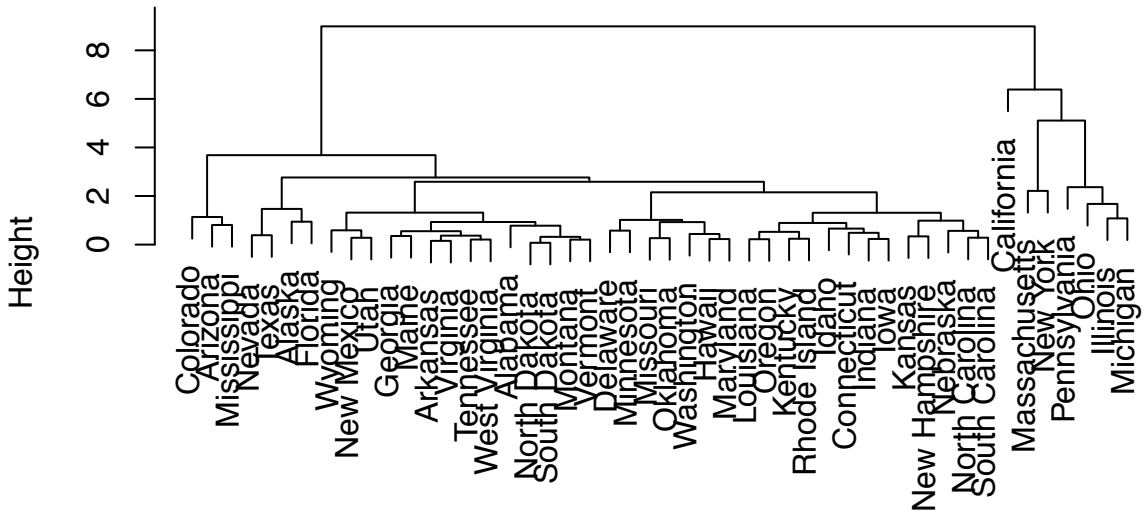

```
# t_slength by salary_real
plot(merged[,2],merged[,5],col=rgb(0,0,merged[,"Cluster"]),pch=16,xlab="t_slength",ylab="expend")
```




```
# salary_real by expend
plot(merged[,4],merged[,5],col=rgb(0,0,merged[,"Cluster"]),pch=16,xlab="salary_real",ylab="expend")
```




```
# All the scatter plots indicate that the data is pretty well clustered across
# the various contionus features that contribute to "Professionalism."
plot(hc_complete, xlab="State")
```




```
# We see from the dendrogram that the states classified as cluster 2
# are immediately seperated from the states classified as cluster 1
# prior to further segmentation on features.
# In essence, the most similar states were clustered together at the start.

# centroids
gmm1$mu
```

```
## [1] -0.3919018  1.1807291
```

```
kmeans$center
```

```
##    t_slength    slength salary_real     expend
## 1  2.0079549  2.0643454     2.04323  1.4647791
## 2 -0.2868507 -0.2949065    -0.29189 -0.2092542
```

```
# For the gmm model, the centroid for cluster 1 is around -0.4
# and the centroid for cluster 2 is around 1.2.
# meanwhile, the kmeans clustering algo designates the centroid
# for cluster 1 to be around 2 and the centroid for cluster 2 to
# be around -0.3.

# Please note that for whatever reason the comparative clusters
# swapped. That is, cluster 1 from kmeans is most similar to
# cluster 2 from gmm in terms of content (please ignore the labeling relation).

# From a more abstract perpective, the models (hc, gmm, kmeans) agree on both the number
# and specific states that should be clustered togeter, as well as roughly
# where the centroids should be designated (with some difference) (for kmeans and gmm).
```

Problem 8

```
library(clValid)
```

```
## Loading required package: cluster
```

```
library(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
```

```
rownames(df) <- 1:nrow(df)
intern <- clValid(df, 2, clMethods = c("hierarchical", "kmeans","model"),
                  validation = "internal", method = "complete")
summary(intern)
```

```
##
## Clustering Methods:
##  hierarchical kmeans model
##
## Cluster sizes:
##  2
##
## Validation Measures:
##                                  2
##
## hierarchical Connectivity   8.1250
##              Dunn           0.1674
##              Silhouette     0.6144
## kmeans       Connectivity   8.5683
##              Dunn           0.1726
##              Silhouette     0.6390
## model        Connectivity  18.7095
##              Dunn           0.0833
##              Silhouette     0.4230
##
## Optimal Scores:
##
##              Score  Method       Clusters
## Connectivity 8.1250 hierarchical 2
## Dunn         0.1726 kmeans       2
## Silhouette   0.6390 kmeans       2
```

```
# The Silhouette Width and Dunn Index score both measure compactness
# and seperation of the clusters. The Silhouette value measures the
# degree of confidence in a particular clustering assignment in which
# well-clustered data produces a value closer to 1 and poorly clustered
# data produces a value closer to -1. The Dunn Index value lies between
# 0 and infinity and should be maxed (in the case of well-clustered data)
```

```r
# From the summary information, we see hierarchical produce a Dunn value
# of approx. 0.17 and a silhouette value of approx. 0.61.
# Kmeans produces Dunn = 0.17 and Silhouette = 0.64.
# Gmm produces Dunn = 0.08 and Silhouttte = 0.42.
# As a result, the kmeans method seems to be optimal for clustering our data.

# In fact, even when we look at optimal scores, kmeans is the best
# in terms of it's Dunn Index and Silhouette values. It is important to note,
# however, that the hierarchical algorithm is better than kmeans when assessing
# clustering power on the Connectivity value.

# In none of the cases is the gmm algorithm the best.

# Now lets check for different levels of k.

intern2 <- clValid(df, c(2:5,10), clMethods = c("hierarchical", "kmeans","model"),
                   validation = "internal", method = "complete")
summary(intern2)
```

```
##
## Clustering Methods:
##  hierarchical kmeans model
##
## Cluster sizes:
##  2 3 4 5 10
##
## Validation Measures:
##                                  2       3       4       5      10
##
## hierarchical Connectivity   8.1250 10.7417 13.1345 18.7563 45.0500
##              Dunn           0.1674  0.2093  0.2902  0.2836  0.2515
##              Silhouette     0.6144  0.5820  0.5199  0.3943  0.3275
## kmeans       Connectivity   8.5683 17.7806 18.1651 21.6810 45.0500
##              Dunn           0.1726  0.1671  0.2456  0.1214  0.2515
##              Silhouette     0.6390  0.5047  0.4824  0.3495  0.3275
## model        Connectivity  18.7095 23.7964 33.3683 60.1651 62.1766
##              Dunn           0.0833  0.0855  0.0554  0.0280  0.0928
##              Silhouette     0.4230  0.3854  0.2157  0.0962  0.2132
##
## Optimal Scores:
##
##              Score  Method       Clusters
## Connectivity 8.1250 hierarchical 2
## Dunn         0.2902 hierarchical 4
## Silhouette   0.6390 kmeans       2
```

```r
# Under k = 2, kmeans performed best under Dunn and Silhouette.
# Similarly, kmeans at k = 4 performed better, according to Dunn,
# than kmeans at k = 2. However, according to Silhouette, kmeans
# with k = 2 is still optimal. Meanwhile, kmeans' connectivity is
# minimized (the goal), under k = 2 as well. Further examination is
# needed before confidently selecting  an "optimal" value for k.
```

Problem 9

```
# From our results, we see that there is no perfect fit that is better
# than the other algorithms or k-values across the entire board.
# Selecting one algorithm at one value of k means discounting evidence
# pointing to a better fit elsewhere.
# For example, if we chose kmeans with k = 2, we would be discounting
# the evidence of a more optimal connectivity value with hierarchical
# and also evidence of a higher dunn value for kmeans at k = 4.
# That said, we are still optimizing Dunn compared to other algorithms at
# the same k level as well as Silhouette value.
# In essence, there is no entirely "optimal" approach. As such, further
# exploration is required and a slightly less than optimal model might
# be implemented.

# There are many times when a sub-optimal clustering method could be
# used regardless of validation statistics.
# 1. Clustering noise
#   A clustering algorithm with great validation statistics might
#   be clustering the noise in the data together with the actually
#   valuable data itself. For example, in clustering financial metrics
#   that contribute to a stock price's increase, it is important to
#   understand that the real world experiences plenty of events that
#   contribute to "white noise" in financial data - pure coincidences
#   that don't contribute to an overall pattern.
# 2. Simplicity/interpretability
#   a clustering algorithm that is too complex might also be difficult
#   to interpret. An algo that maps individual's preferred flavor profile
#   in ice cream, for example, might lack human interpretability if the algo
#   is made too complex.
# 3. Bias/Variance Trade-off
#   As a model increases in complexity (number of clusters designated), the
#   model will have little to no bias but very high variance. This tradeoff
#   implies that the model is overfitting the data and is probably not robust
#   enough to handle noise in the data with accuracy.
```