PS3 - LR

knitr::opts\_chunk$set(echo = TRUE)

## 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:

#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 wmyith our discussion of these data and the concept of state legislative professionalism from class for relevant background information.  
  
load("/Users/lillyreich/Downloads/legprof-components.v1.0.RData")

#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);

xsub <- subset(x, select = -c(fips,stateabv,sessid,mds1,mds2))

#b) restrict the data to only include the 2009/10 legislative session for consistency;

xsubyr <- xsub[ which(xsub$year=='2009'  
 | xsub$year=='2010'),]

#c) omit all missing values;

xsubyr\_mod <- xsubyr[ complete.cases(xsubyr), ]  
state\_names <- subset(xsubyr\_mod, select = c(state))  
xsubyr\_mod <- subset(xsubyr\_mod, select = -c(year,state))

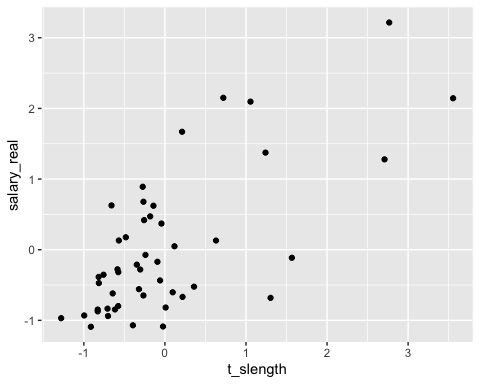
#d) standardize the input features;

#scaled\_xmod <- xsubyr\_mod  
#scaled\_xmod[, -c(year,state)] <- scale(scaled\_xmod[, -c(year,state)])  
#summary(scaled\_xmod)  
  
#xsubyr$year <- NULL  
#xsubyr$state <- NULL  
  
xsubyr\_mod <- data.frame(scale(xsubyr\_mod))  
  
xsubyr <- xsubyr\_mod

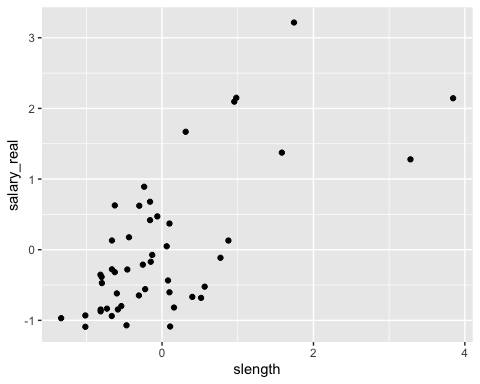
#e) and anything else you think necessary to get this subset of data into workable form (hint: consider storing the state names as a separate object to be used in plotting later)

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

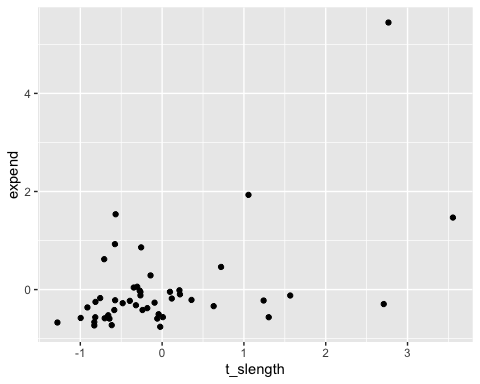
library(ggplot2)  
  
ggplot(xsubyr) +   
 geom\_point(mapping = aes(x = t\_slength, y = salary\_real))



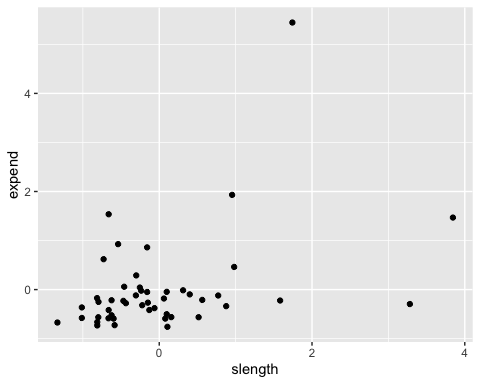
#The scatter plot displays a positive relationship between total length of session  
# and the real salary of legislator for the years of 2009 and 2010. However, the trend   
# isn't very clear visually.  
ggplot(xsubyr) +   
 geom\_point(mapping = aes(x = slength, y = salary\_real))



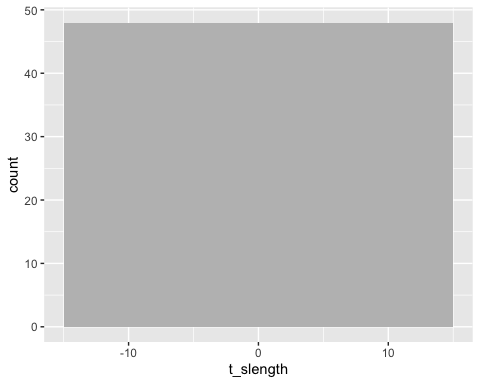
#The scatter plot displays a positive relationship between length of regular session  
# and the real salary of legislator for the years of 2009 and 2010. However, the trend   
# isn't very clear visually.  
ggplot(xsubyr) +   
 geom\_point(mapping = aes(x = t\_slength, y = expend))



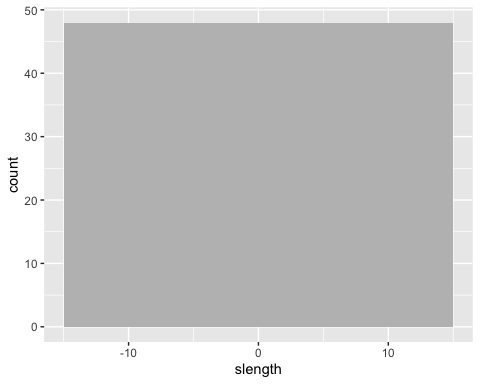
#The scatter plot doesn't display a clear trend. There are outliers in the dataset  
ggplot(xsubyr) +   
 geom\_point(mapping = aes(x = slength, y = expend))



#The scatter plot doesn't clearly display a trend. There are outliers in the dataset  
ggplot(xsubyr, mapping = aes(x = t\_slength)) +  
 geom\_histogram(binwidth = 30.0, fill = "grey")

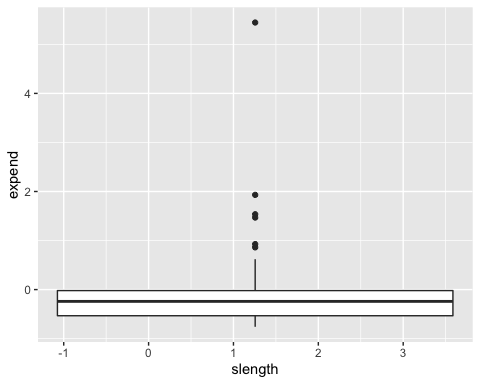


#The histogram displays somewhat of a skewed normal distribution of total session length around   
# a mean of 130. There are significant outliers  
ggplot(xsubyr, mapping = aes(x = slength)) +  
 geom\_histogram(binwidth = 30.0, fill = "grey")



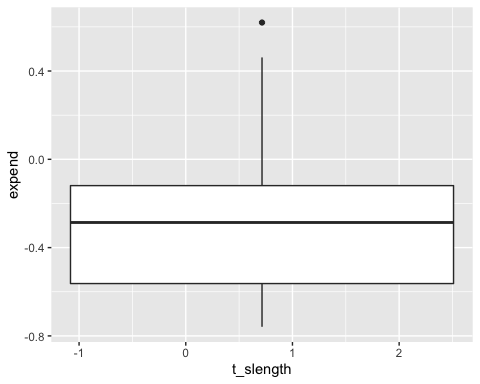
#The histogram displays somewhat of a skewed normal distribution of total session length around   
# a mean of 130. There are significant outliers. The similarity between slength and t\_slength   
# indicates the small effect of special sessions on the distribution  
ggplot(xsubyr, mapping = aes(x = slength, y=expend)) +  
 geom\_boxplot()

## Warning: Continuous x aesthetic -- did you forget aes(group=...)?

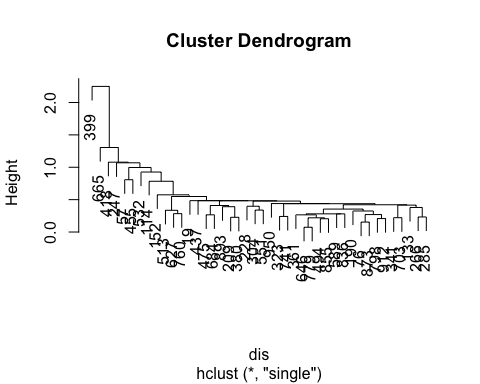


outliers <- boxplot(xsubyr$expend, plot=FALSE)$out  
miout <- xsubyr  
miout <- miout[-which(miout$expend %in% outliers),]  
ggplot(miout, mapping = aes(x = t\_slength, y=expend)) +  
 geom\_boxplot()

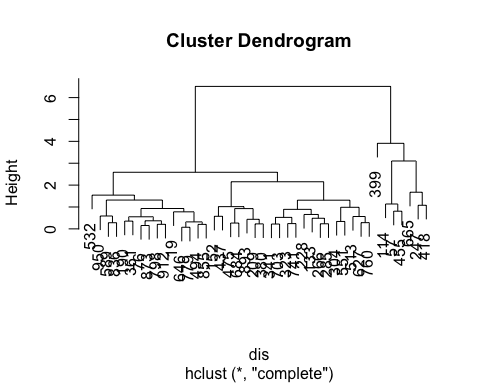
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?

 #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.

dis <- dist(miout, method = 'euclidean')  
clus\_single <- hclust(dis, method="single")  
plot(clus\_single)



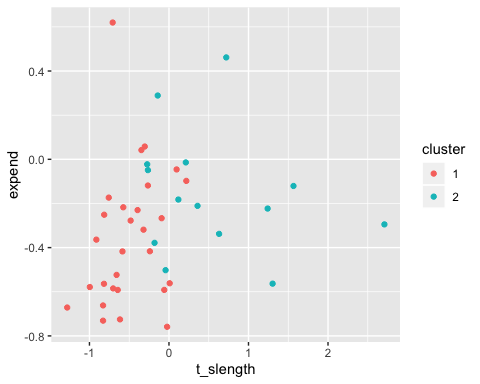
clus\_comp <- hclust(dis, method="complete")  
plot(clus\_comp)



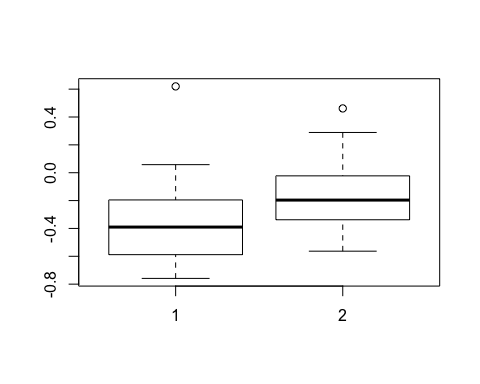
#The cluster dendograms indicate non-random structure or clustering in the dataset.

#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.

#install.packages("cluster",repos="http://cran.us.r-project.org")  
library(cluster)  
kmean\_clust <- kmeans(miout, 2)  
miout\_km <- miout  
miout\_km$cluster <- as.factor(kmean\_clust$cluster)  
ggplot(miout\_km) +   
 geom\_point(mapping = aes(x = t\_slength, y = expend, color = cluster))



#From this reiterated scatter plot of the total session length versus expenditure, we can clearly   
#see the data is separated between two clusters.   
boxplot(miout\_km$expend ~ miout\_km$cluster)



#The above boxplot shows two means for the clustered data.

#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.

installed.packages("plotGMM")

## Package LibPath Version Priority Depends Imports LinkingTo Suggests  
## Enhances License License\_is\_FOSS License\_restricts\_use OS\_type Archs  
## MD5sum NeedsCompilation Built

installed.packages("dplyr")

## Package LibPath Version Priority Depends Imports LinkingTo Suggests  
## Enhances License License\_is\_FOSS License\_restricts\_use OS\_type Archs  
## MD5sum NeedsCompilation Built

installed.packages("mixtools")

## Package LibPath Version Priority Depends Imports LinkingTo Suggests  
## Enhances License License\_is\_FOSS License\_restricts\_use OS\_type Archs  
## MD5sum NeedsCompilation Built

library(plotGMM)  
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

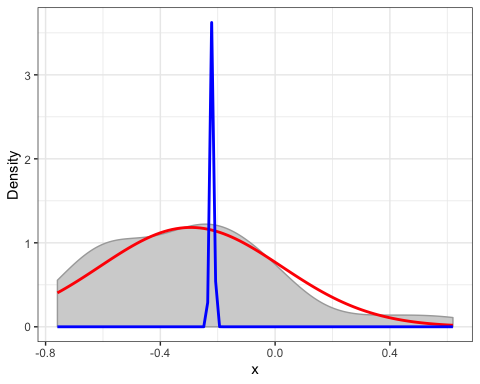
library(mixtools)

## mixtools package, version 1.1.0, Released 2017-03-10  
## This package is based upon work supported by the National Science Foundation under Grant No. SES-0518772.

mixmdl <- mixtools::normalmixEM(miout$expend, k = 2)

## number of iterations= 72

plot\_GMM(mixmdl,2)

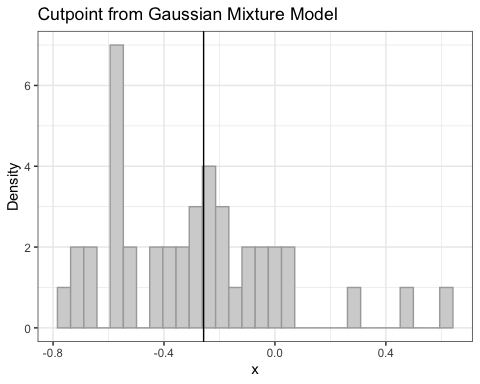


plot\_cut\_point(mixmdl, plot = TRUE)

## Warning in if (color == "amerika") {: the condition has length > 1 and only  
## the first element will be used

## Warning in if (color == "wesanderson") {: the condition has length > 1 and  
## only the first element will be used

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



plot\_cut\_point(mixmdl, plot = FALSE)

## [1] -0.257183

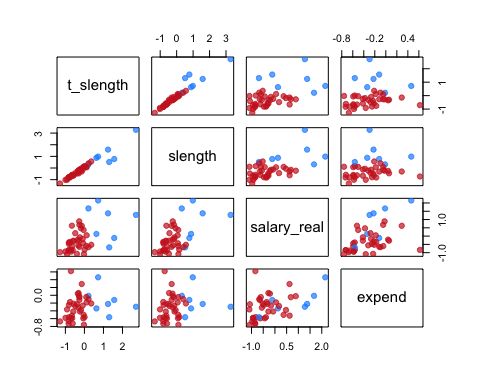
# The dataset was soft partitioned using GMM with the cutoff point at 378 for expenditure.  
# The dataset seems to be partitioned well when visually examined and falls approximately in   
# two normal distributions.

#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.

installed.packages("ppclust")

## Package LibPath Version Priority Depends Imports LinkingTo Suggests  
## Enhances License License\_is\_FOSS License\_restricts\_use OS\_type Archs  
## MD5sum NeedsCompilation Built

library(ppclust)  
myfcm <- fcm(miout, centers=2)  
plotcluster(myfcm, cp=1, trans=TRUE)



as.data.frame(myfcm$u)

## Cluster 1 Cluster 2  
## 19 0.05844213 0.9415579  
## 57 0.82954291 0.1704571  
## 76 0.03656678 0.9634332  
## 114 0.89663270 0.1033673  
## 133 0.35398170 0.6460183  
## 152 0.21636121 0.7836388  
## 190 0.04351342 0.9564866  
## 209 0.44002198 0.5599780  
## 228 0.14721049 0.8527895  
## 247 0.76745009 0.2325499  
## 266 0.10338730 0.8966127  
## 285 0.09547102 0.9045290  
## 304 0.17449006 0.8255099  
## 323 0.03152063 0.9684794  
## 341 0.04967206 0.9503279  
## 361 0.02345441 0.9765456  
## 380 0.39372643 0.6062736  
## 399 0.73532442 0.2646756  
## 418 0.80846654 0.1915335  
## 437 0.09729134 0.9027087  
## 455 0.63478537 0.3652146  
## 475 0.42403822 0.5759618  
## 494 0.05092156 0.9490784  
## 513 0.20158635 0.7984136  
## 532 0.13368487 0.8663151  
## 551 0.18172960 0.8182704  
## 589 0.08752002 0.9124800  
## 627 0.48996102 0.5100390  
## 646 0.06263920 0.9373608  
## 665 0.89962652 0.1003735  
## 684 0.34985734 0.6501427  
## 703 0.06726369 0.9327363  
## 741 0.01890240 0.9810976  
## 760 0.34407741 0.6559226  
## 779 0.06569871 0.9343013  
## 798 0.01503105 0.9849690  
## 836 0.08494704 0.9150530  
## 855 0.04808094 0.9519191  
## 873 0.03941384 0.9605862  
## 893 0.39743262 0.6025674  
## 912 0.01851243 0.9814876  
## 950 0.13010445 0.8698956

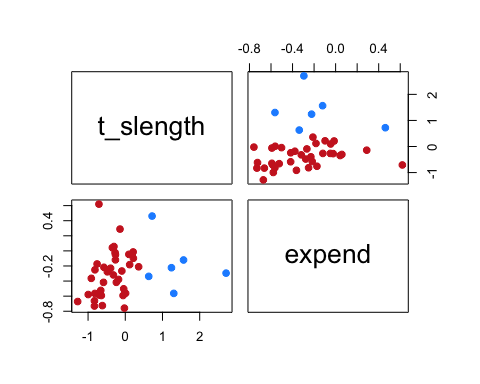
summary(myfcm)

## Summary for 'myfcm'  
##   
## Number of data objects: 42   
##   
## Number of clusters: 2   
##   
## Crisp clustering vector:  
## [1] 2 1 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 1 1 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2  
## [36] 2 2 2 2 2 2 2  
##   
## Initial cluster prototypes:  
## t\_slength slength salary\_real expend  
## Cluster 1 0.7203749 0.9806336 2.1502555 0.4616150  
## Cluster 2 -0.2650671 -0.3059914 -0.6474221 -0.1186464  
##   
## Final cluster prototypes:  
## t\_slength slength salary\_real expend  
## Cluster 1 0.8331062 0.8764421 0.6783639 -0.1527111  
## Cluster 2 -0.4559783 -0.4446071 -0.4536309 -0.3395177  
##   
## Distance between the final cluster prototypes  
## Cluster 1  
## Cluster 2 4.723219  
##   
## Difference between the initial and final cluster prototypes  
## t\_slength slength salary\_real expend  
## Cluster 1 0.1127314 -0.1041915 -1.4718917 -0.6143261  
## Cluster 2 -0.1909112 -0.1386157 0.1937912 -0.2208712  
##   
## Root Mean Squared Deviations (RMSD): 1.163921   
## Mean Absolute Deviation (MAD): 6.09466   
##   
## Membership degrees matrix (top and bottom 5 rows):   
## Cluster 1 Cluster 2  
## 19 0.05844213 0.9415579  
## 57 0.82954291 0.1704571  
## 76 0.03656678 0.9634332  
## 114 0.89663270 0.1033673  
## 133 0.35398170 0.6460183  
## ...  
## Cluster 1 Cluster 2  
## 855 0.04808094 0.9519191  
## 873 0.03941384 0.9605862  
## 893 0.39743262 0.6025674  
## 912 0.01851244 0.9814876  
## 950 0.13010445 0.8698956  
##   
## Descriptive statistics for the membership degrees by clusters  
## Size Min Q1 Mean Median Q3 Max  
## Cluster 1 7 0.6347854 0.7513873 0.7959755 0.8084665 0.8630878 0.8996265  
## Cluster 2 35 0.5100390 0.7910262 0.8435282 0.9045290 0.9511235 0.9849690  
##   
## Dunn's Fuzziness Coefficients:  
## dunn\_coeff normalized   
## 0.7625822 0.5251644   
##   
## Within cluster sum of squares by cluster:  
## 1 2   
## 17.26499 25.11236   
## (between\_SS / total\_SS = 39.4%)   
##   
## Available components:   
## [1] "u" "v" "v0" "d" "x"   
## [6] "cluster" "csize" "sumsqrs" "k" "m"   
## [11] "iter" "best.start" "func.val" "comp.time" "inpargs"   
## [16] "algorithm" "call"

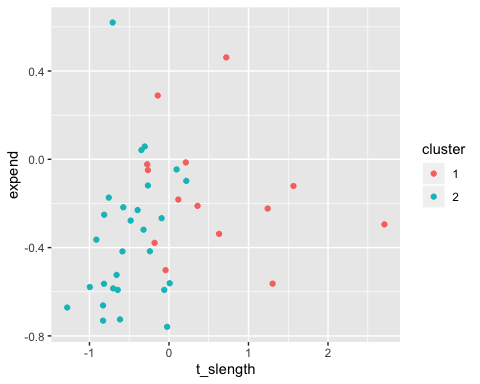
#The plots display the fuzzy c-means algorithm's results. Clearly the   
# data is partitioned appropriately for the expenditure variable

#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).

miout\_sub\_jstexp <- subset(miout, select = -c(slength,salary\_real))  
pltfcm <- fcm(miout\_sub\_jstexp, centers=2)  
plotcluster(pltfcm)



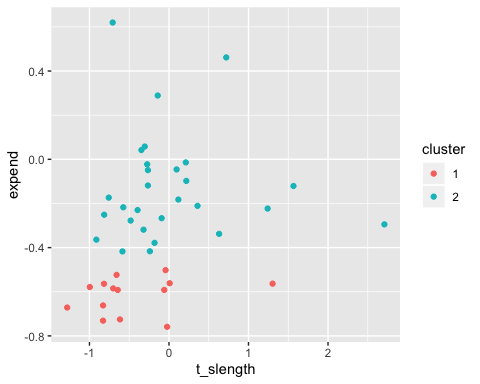
km\_sol <- kmeans(miout, 2)  
miout\_sub <- miout  
miout\_sub$cluster <- as.factor(km\_sol$cluster)  
ggplot(miout\_sub) +   
 geom\_point(mapping = aes(x = t\_slength, y = expend, color = cluster))



mixmdl <- mixtools::normalmixEM(miout$expend, k = 2)

## number of iterations= 21

posterior <- data.frame(cbind(mixmdl$x, mixmdl$posterior))  
posterior$component <- ifelse(posterior$comp.1 > 0.3, 1, 2)  
miout\_sub <- miout  
miout\_sub$cluster <- as.factor(posterior$component)  
ggplot(miout\_sub) +   
 geom\_point(mapping = aes(x = t\_slength, y = expend, color = cluster))



#From this reiterated scatter plot of the total session length versus expenditure, we can clearly   
#see that the data is separated between two clusters. This is true for all three methods.

#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).

installed.packages("clValid")

## Package LibPath Version Priority Depends Imports LinkingTo Suggests  
## Enhances License License\_is\_FOSS License\_restricts\_use OS\_type Archs  
## MD5sum NeedsCompilation Built

library(clValid)  
installed.packages("purrr")

## Package LibPath Version Priority Depends Imports LinkingTo Suggests  
## Enhances License License\_is\_FOSS License\_restricts\_use OS\_type Archs  
## MD5sum NeedsCompilation Built

library(purrr)  
installed.packages("DiscriMiner")

## Package LibPath Version Priority Depends Imports LinkingTo Suggests  
## Enhances License License\_is\_FOSS License\_restricts\_use OS\_type Archs  
## MD5sum NeedsCompilation Built

library(DiscriMiner)  
k.values <- 1:20  
outliers <- boxplot(xsubyr$expend, plot=FALSE)$out  
miout <- xsubyr  
miout <- miout[-which(miout$expend %in% outliers),]  
wssgmm <- function(k) {  
 mixmdl\_cust = mixtools::normalmixEM(miout$expend, k)  
 posterior <- data.frame(cbind(mixmdl\_cust$x, mixmdl\_cust$posterior))  
 posterior$component <- ifelse(posterior$comp.1 > 0.5, 1, 2)  
 miout\_sub <- miout  
 miout\_sub$cluster <- as.factor(posterior$component)  
 sum(withinSS(miout, posterior$component))  
   
}  
wsskm <- function(k) {  
 kmeans(miout, k, nstart=10)$tot.withinss  
}  
wssfcm <- function(cent) {  
 fcm(miout, centers=cent)$sumsqrs$tot.within.ss  
}  
wss\_km <- map\_dbl(k.values,wsskm)  
cent.values <- 1:20  
wss\_fcm <- map\_dbl(cent.values,wssfcm)  
wss\_gmm <- map\_dbl(k.values,wssgmm)

## number of iterations= 85   
## number of iterations= 44   
## number of iterations= 74   
## number of iterations= 22   
## number of iterations= 75   
## number of iterations= 57   
## number of iterations= 13   
## number of iterations= 94   
## number of iterations= 79   
## number of iterations= 80   
## number of iterations= 52   
## number of iterations= 71   
## number of iterations= 59   
## number of iterations= 30   
## number of iterations= 92   
## number of iterations= 86   
## number of iterations= 69   
## number of iterations= 73   
## number of iterations= 20   
## number of iterations= 10

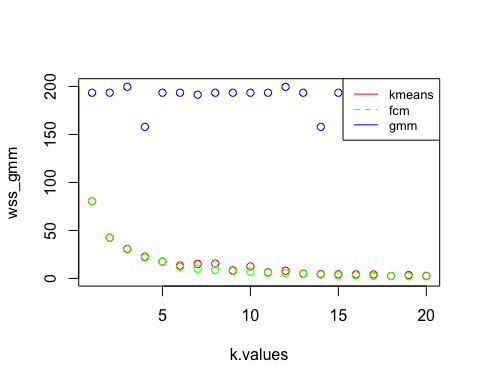
plot(k.values, wss\_gmm, col="blue", ylim=c(0,200))  
points(cent.values, wss\_fcm,col="red")  
print(wss\_fcm)

## [1] 80.477960 42.377351 30.737669 22.603156 17.529474 13.397241 15.076695  
## [8] 15.583598 8.371964 12.462514 6.398583 7.953488 4.938152 4.557304  
## [15] 4.262053 4.200566 4.109735 2.523043 3.540232 2.685648

print(wss\_km)

## [1] 80.477960 42.377351 30.191978 21.728652 17.492421 11.974506 10.030693  
## [8] 8.878638 7.582377 7.147079 5.887968 5.347673 4.550975 3.964287  
## [15] 3.311123 3.144455 2.652537 2.438458 2.352093 2.151307

points(k.values, wss\_km, col="green")  
legend("topright", legend=c("kmeans","fcm","gmm"), col=c("red","green","blue"), lty=1:2, cex=0.8)



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:purrr':  
##   
## map

## The following object is masked from 'package:mixtools':  
##   
## dmvnorm

#s\_mat <- as.matrix(xsubyr[,c(-1,-6)])  
#s\_mat\_na <- na.omit(s\_mat)  
#valid <- clValid(s\_mat\_na,2:10,  
# clMethods = "model")  
#valid  
#call all three methods use c parenthesis and quotes!

#10) Discuss the validation output.

#a. What can you take away from the fit?  
#Since kmeans and fcm have similar results, it might be that soft partitioning was unnecessary  
# kmeans even has a lower wss at a given k if any difference exists.  
# As k increases, this effect becomes more obvious. The WSS from GMM might not be an appropriate analysis technique but it indicates that GMM performs worse than the other two methods.  
# b. Which approach is optimal? And optimal at what value of k?  
# Kmeans is optimal and its optimal for higher values of k as wss decreases with more clustering.  
# Around k = 4 or k = 5, the decrease in wss becomes relatively small for higher values of k.  
# c. What are reasons you could imagine selecting a technically "sub-optimal"  
#partitioning method, regardless of the validation statistics?   
# In some cases, it may be desirable to apply soft or hard partitioning because the physical interpretation  
# of the dataset may suggest this. In other cases, it may be that mixture models or similar approaches are   
# the desired form of solution although their implementation may be suboptimal.

```

## Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.