Unsupervised ML HW#2 -Lilly Reich

Computation

You fielded a survey and collected some wildly descriptive feature vectors. Use the following vectors to address questions 1-3:

p{1, 2} q{3, 4}

#1) Calculate Manhattan, Canberra, and Euclidean distances “by hand” (i.e., create the data, program each line, and make the calculations). What are the values for each measure?

Manhattan: Dist = abs(x1-x2) + abs(y1-y2) = 2 + 2 = 4

Euclidean: Dist = sqrt((x1-x2))^2 + (y1-y2)^2) = sqrt(4+4) = 2\*sqrt(2) = 2.82

Canberra: Dist = abs(x1 -x2) / (x1-x2) + abs(y1-y2)/(y1+y2) = 2/(1+3) + 2/(2+4) = 5/6 = 0.83

#2) Use the dist() function in R to check your work. Were you right or wrong? (be honest in your reporting). If wrong, after debugging, where and why did you go wrong?

x1 <- c(1,2)  
x2 <- c(3,4)  
  
euc <- dist(rbind(x1,x2), method = "euclidean")   
can <- dist(rbind(x1,x2), method = "canberra")   
man <- dist(rbind(x1,x2), method = "manhattan")  
  
  
print("euclidean:")

## [1] "euclidean:"

print(euc)

## x1  
## x2 2.828427

print("canberra:")

## [1] "canberra:"

print(can)

## x1  
## x2 0.8333333

print("manhattan:")

## [1] "manhattan:"

print(man)

## x1  
## x2 4

I was correct with the distance calculations by hand.

#3) What are the key differences between these measures, and why does it matter? How might you see these differences “in action” with these fictious data?

The key differences between Euclidean and Manhattan is that Manhattan is raised to a lower power (1 vs.2) which gives more significance to smaller deviations. Therefore, this becomes important in high dimensionalities.

In the results above, we can notice the higher Manhattan distance of 4 versus the lower 2.83.

Canberra provides higher significance to values closer to zero since the distance is normalized by the absolute sum of the values and in turn gives lower significance to higher distance (>1). Therefore, this becomes important in identifying deviations from normal distributions.

From the data, we may notice that Canberra has a reduced distance due to the normalization by division of absolute values.

Download the old faithful data set (<https://stat.ethz.ch/R-manual/Rpatched/library/datasets/html/faithful.html>), and use to address questions 4-6:

#4) Use some basic EDA techniques to present and discuss the data (e.g., visualize, describe in multiple ways, etc.)

#install.packages("seriation")  
library(seriation)  
#install.packages("DataExplorer")  
library(DataExplorer)  
#install.packages("plotly")  
library(plotly)

## Loading required package: ggplot2

##   
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

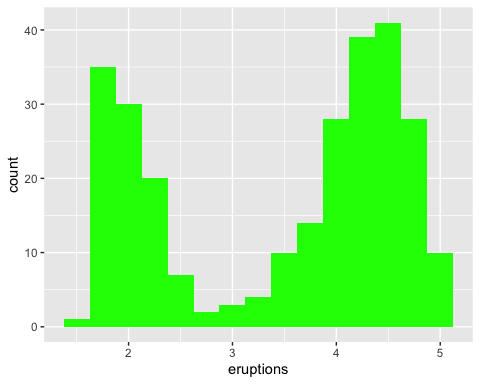
## The following object is masked from 'package:stats':  
##   
## filter

## The following object is masked from 'package:graphics':  
##   
## layout

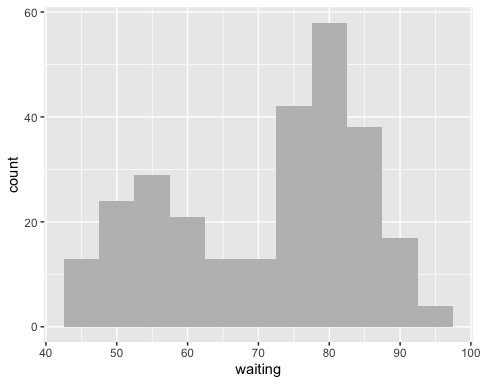
data("faithful")  
df <- faithful  
head(df)

## eruptions waiting  
## 1 3.600 79  
## 2 1.800 54  
## 3 3.333 74  
## 4 2.283 62  
## 5 4.533 85  
## 6 2.883 55

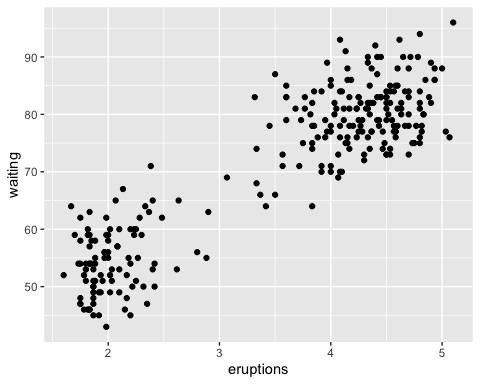
#install.packages("ggplot2")  
library(ggplot2)  
  
ggplot(df, mapping = aes(x = eruptions)) +   
 geom\_histogram(binwidth = 0.25, fill = "green")



#This histogram of eruption data indicates there are two   
#independent distributions; the first between eruptions heights of   
#approximately 1.7 to 2.7 and the other approximately between 3.2 to 5.1.   
#The distributions almost appear normal.   
  
ggplot(df, mapping = aes(x=waiting)) + geom\_histogram(binwidth = 5.0 , fill = "grey")

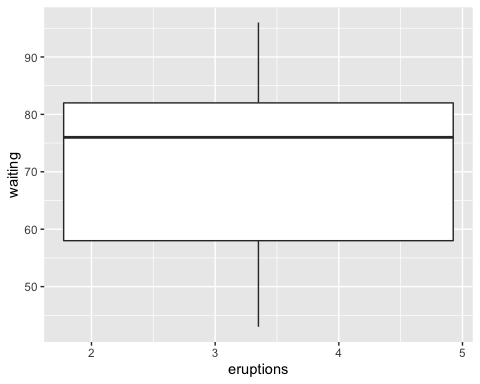


#Plotting the histogram for the waiting times indicates similiar trends as   
#observed, above.   
  
ggplot(df) + geom\_point(mapping=aes(x=eruptions, y = waiting))



#This scatter plot displays the waiting time and length of eruptions  
#as the y and x axis, respectively. Therefore, this reveals higer eruptions  
# are associated with longer wait times.   
  
ggplot(df, aes(x=eruptions, y =waiting)) +   
 geom\_boxplot()

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



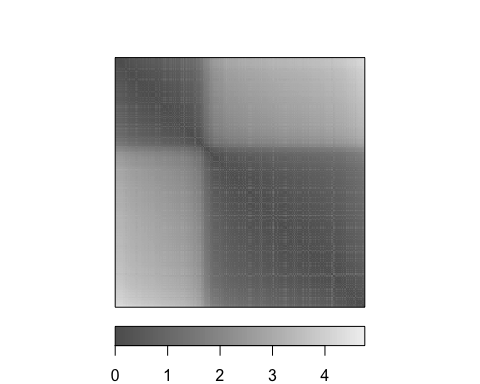
#This box plot displays a mean around 76 for waiting time; therefore, most of   
#the data is clustered around the higher values. We may also notice there are   
#no outliers in the data.

#5) Calculate a dissimilarity matrix of these data.

#install.packages("cluster")  
library(cluster)  
df <- daisy(faithful, metric = "euclidean", stand = FALSE)

#6) Generate an ODI for the Old Faithful data. What do you see?

library("seriation")  
# faithful data: ordered dissimilarity image  
df\_scale<- scale(faithful)  
df\_dist <- dist(df\_scale)   
dissplot(df\_dist)



#In this figure, we may notice the data was originally a dissimilarity matrix that   
#was eventually calculated by a Euclidean distance to receive the   
#nearest measure per cluster. The figure also suggests there are two clusters in the Old   
#Faithful dataset since there is a cluster on the upper left and the lower left.

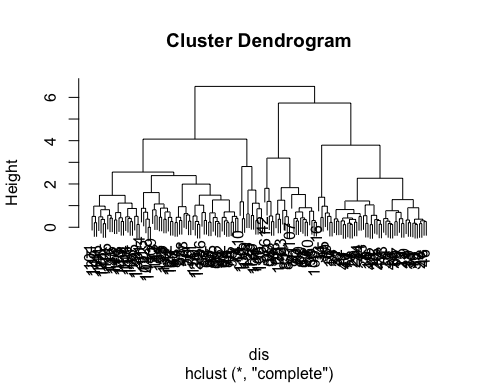
Download the Iris data set we used in class (e.g., data(iris) in R), and use to address questions 7-10:

#7) Using any munging tools you’d like (e.g., dplyr from the Tidyverse), create a subset of the data excluding the species feature, scaling the features, and calculating a dissimilarity matrix (think “pipe” for stacking functions to do this quickly, e.g.).

data(iris)  
df <- iris   
df <- subset(df,select = -c(Species))  
df\_scl <- scale(df)  
dis <- dist(df\_scl, method = 'euclidean')

#8) Fit an agglomerative hierarchical clustering algorithm using complete linkage on your subset data and render the dendrogram of clustering results. What do you see?

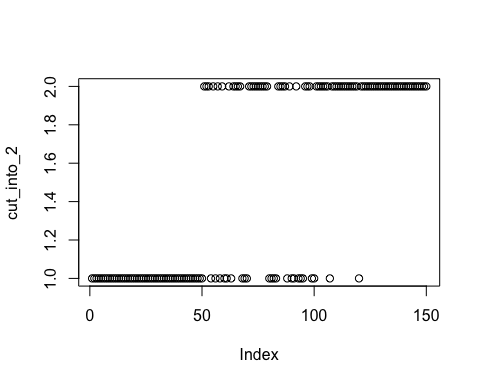
clus <- hclust(dis, method = "complete")  
plot(clus)



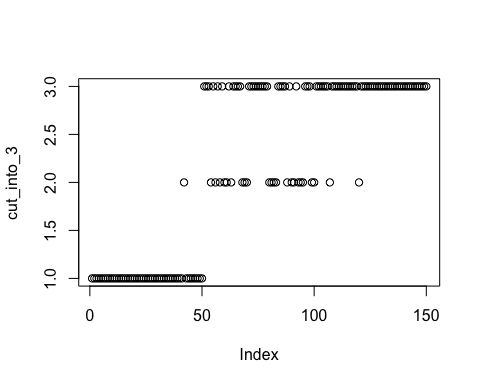
#It seems the dataset fits into 4 clusters at a distance of 4. We may also notice  
#further clustering occurs at a distance of 5 with two remaining clusters and   
#eventually the data fits in one cluster a distance of 6.

#9) Try cutting the tree at 2 and 3 and show these trees side-by-side. How do they differ?

cut\_into\_2 <-cutree(clus, k = 2)   
cut\_into\_3 <-cutree(clus, k = 3)  
plot(cut\_into\_2)



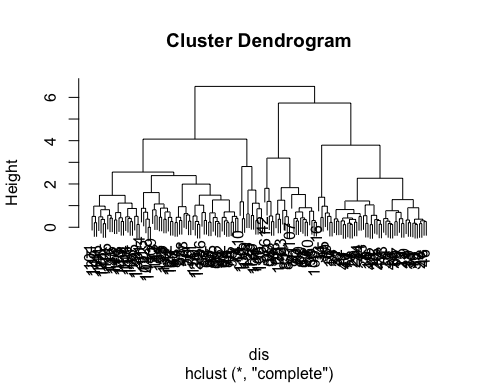
plot(cut\_into\_3)



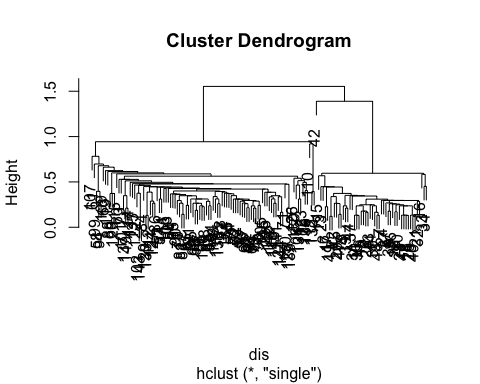
#The cutree with 3 clusters seems to take most of its members from the 1st   
#cluster in the 2-cluster cutree. In the cut\_into\_3 plot, the data is further  
#divided according to distance when compared to cut\_into\_2.

#10) Now fit the algorithm using single and complete linkage and present each dendrogram side-by-side. Discuss the differences. What effects can we see in the clustering patterns when using different linkage methods?

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



clus\_single <-hclust(dis, method = "single")  
plot(clus\_single)



# In Complete Linkage Hierarchical Clustering, the complete-linkage is done by  
#using the maximum pairwise distance between two clusters. Therefore, we see  
#delayed clustering and more clusters when we compare it to Single Linkage  
#Hierarchical Clustering with single linkage. Single Linkage Hierarchical   
#Clustering is done by the minimum pairwise distance between two clusters (which  
#is a less stringent requirement for clustering analysis).

#Critical thinking

#1) You just assessed the clusterability of some feature space, ℝ. #Address the following questions:

#a. How would you go about determining whether clustering made sense #to consider or not?

#The cluster dendrogram would reveal if the data set needed a clustering analysis or not. The magnitude of the distance (or height) would also indicate if the clustering is done according to random fluctuation or actual differences in the data points of different clusters.

#It is also necessary to consider whether the data’s physical attributes indicate clustering. For example, it may be known that the data behaves differently at different input parameters(e.g. day hours versus evening hours).

#b. What are techniques you would use, and what might you be looking for from each?

#A scatter plot or a histogram of the data would vaguely display clustering of certain data points and might be a revealing EDA (Exploratory Data Analysis) technique. If the data is somehow agglomerated around certain values in different regions on the plot,then clustering would be useful.

#Also, a revealing statistic may be the Hopkins Statistic. A value close to one would indicate the need for clustering.

#We may also want to replicate the data collection and if similiar clustering is observed, then the clustering analysis is revealing actual trends.

#We could also use several clustering techniques and if their results were similiar then the analysis is indeed valid.

#c. How might these techniques work together to motivate clustering or not?

#The more trends observed from these multiple analyses, the more valid the clustering analysis results will become.

#d. And ultimately, can/should you proceed if you find little to no support for clusterability? Why or why not?

#No, because any dataset could potentially be clustered by random sampling errors.

#2) Locate (and read) a paper that applies the hierarchical agglomerative clustering technique. Address the following questions:

#a. Describe the author(s) process.

#Zhou, C., & Lung, C.-H. (2008). Application and Evaluation of Hierarchical #Agglomerative Clustering in Wireless Sensor Networks.  
#Lecture Notes in Electrical Engineering Sensor and Ad Hoc Networks, #255–276. doi: 10.1007/978-0-387-77320-9\_13

#In this article, the authors present their Wireless Sensor Networks data #analysis with Hierarchical Agglomerative Clustering  
#using qualitative and quantitative data.

#They first compute so-called resemblance coefficients which are similarity #and dissimilarity coefficients. For the quantitative data, they use Euclidean #distances. For the qualitative data, they compute the Jaccard, Simple #Matching and Sorenson Coefficients.

#For the HAC process, they apply single linkage, complete linkage, #un-weighted pair-group and weighted pair-group methodologies. This is #done by clustering using pre-configured threshold values that are used to #define meaningful differences.

#b. Do they go through similar steps as we covered this week both in setting #the stage for clustering (e.g., assessing clusterability, #calculating distance, etc.), as well as in fitting the algorithm? #If not, what did they omit and does this omission impact their findings #in your opinion?

#The authors don’t necessarily prove that the data is cluster-able; #however, they cluster it using the same techniques we performed, i.e. #by computing distance matrices. It’s possible that their lack of #assessment of clusterability may result in meaningless clustering. #However, the authors used metrics for meaningful differences from theory.

#c. Describe at least one possible extension from the study that could #emerge based on their findings.

#This study proved the efficacy of hierarchical agglomerative clustering #in the clustering of Wireless Sensor Networks. This could be used to #evaluate the performance of energy efficiency. The protocols used can #also be compared for efficacy with other clustering protocol as well as #different weights to differentiate clusters. ```