Assignment 4

Megan Marziali

Feb 3, 2021

## Part 1: Implementing a Simple Prediction Pipeline

### Problem set-up

##### Loading required packages

The following code chunk loads the required packages for the assignment.

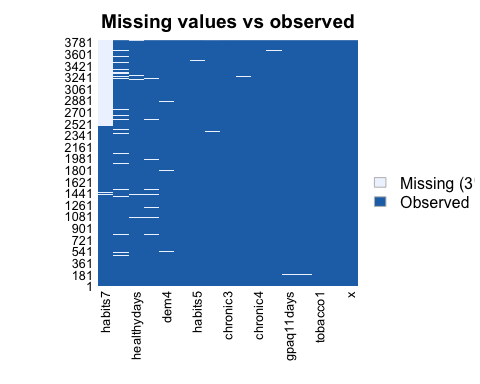
library(tidyverse)  
library(Amelia)  
library(caret)  
library(stats)  
library(factoextra)  
library(cluster)  
  
set.seed(100)

##### Loading data into environment and cleaning

nyc\_data =   
 read.csv("./data/class4\_p1.csv", na = c("", ".", "NA", ".d", ".r")) %>%   
 janitor::clean\_names() %>%  
 mutate(  
 bmi = as.numeric(bmi),  
 gpaq8totmin = as.numeric(gpaq8totmin),  
 gpaq11days = as.integer(gpaq11days),  
 healthydays = as.integer(healthydays),  
 chronic1 = as.factor(chronic1),  
 chronic3 = as.factor(chronic3),  
 chronic4 = as.factor(chronic4),  
 tobacco1 = as.factor(tobacco1),  
 alcohol1 = as.factor(alcohol1),  
 agegroup = as.factor(agegroup),  
 dem3 = as.factor(dem3),  
 dem4 = as.factor(dem4),  
 dem8 = as.factor(dem8),  
 povertygroup = as.factor(povertygroup)  
 )   
  
summary(nyc\_data)

To clean the data, I recoded variables to be factor or numerical variables. The total dataset is comprised of 3811 observations, and 17 variables.

missmap(nyc\_data, main = "Missing values vs observed")



To further explore the data, I mapped the number of missing observations. Based on missingness, I would not include the variables “habits7” and “povertygroup” in any models. I would also remove all missingness from variables that I intend to keep in models.

nyc\_restr =   
 select(nyc\_data, -habits7, -povertygroup) %>%   
 na.omit() %>%   
 distinct(x, .keep\_all = TRUE)  
  
nrow(nyc\_restr)

The variable “habits7” has 1335 missing observations (35.0%) and the variable “povertygroup” has 244 missing observations (6.4%). As both have >5% missing values, I opted to drop them both from the dataset.After removing the variables “habits7” and “povertygroup”, and deleting missing observations from all other included variables, the total N for the dataset is 3552.

##### Data partitioning

train.indices = createDataPartition(y = nyc\_restr$healthydays,p = 0.7,list = FALSE)  
  
training = nyc\_restr[train.indices,]  
testing = nyc\_restr[-train.indices,]

### Problem 1:

The code below fits two prediction models.

model\_1 <- lm(healthydays ~ chronic1 + chronic3 + chronic4 + bmi + gpaq8totmin + gpaq11days + habits5 + dem3, data = training)  
summary(model\_1)  
  
model\_2 <- lm(healthydays ~ tobacco1 + alcohol1 + bmi + gpaq8totmin + gpaq11days + habits5 + dem3, data = training)  
summary(model\_2)

Model 1 is a linear regression model, and includes the following features: chronic1 (hypertension: yes vs. no), chronic3 (diabetes: yes vs. no), chronic4 (asthma: yes vs. no), BMI (continuous), gpaq8totmin (minutes of physical activity on chores: continuous), gpaq11days (walking days: continuous), habits5 (physically active: very active, somewhat active, not very active, not active at all), and dem3 (sex: male vs. female).

Model 2 is a linear regression model, and includes the following features: tobacco1 (days smoking 3+ cigarettes: most days, some days, never), alcohol1 (days drinking 2+ drinks: most days, some days, never), BMI (continuous), gpaq8totmin (minutes of physical activity on chores: continuous), gpaq11days (walking days: continuous), habits5 (physically active: very active, somewhat active, not very active, not active at all), and dem3 (sex: male vs. female).

### Problem 2:

The code below applies both models to the testing data and uses mean squared error as the evaluation metric.

fitted\_1 = predict(model\_1,testing,type = 'response')   
fitted\_2 = predict(model\_2,testing,type = 'response')   
observed = testing$healthydays  
  
mse\_1 = mean((observed - fitted\_1)\*\*2)  
mse\_2 = mean((observed - fitted\_2)\*\*2)

The outcome we are investigating is continuous, so the appropriate evaluation metric is to assess mean squared error. As model 1 has the smaller mean squared error, we can determine that model 1 is the preferred prediction model.

### Problem 3:

The implementation of this model would be helpful for preventative care. We could use predictive modeling to identify factors to intervene on, in order to increase the number of healthy days experienced. As the model works to identify predictors of a higher number of healthy days, we know that those predictors function to increase healthy days experienced.

## Part II

### Problem set-up

#### Importing and scaling data

data("USArrests")  
head(USArrests)  
  
#Removing missing  
us\_arrest = na.omit(USArrests)  
  
colMeans(us\_arrest, na.rm = TRUE)  
apply(us\_arrest, 2, sd, na.rm = TRUE)  
  
scale(us\_arrest, center = TRUE, scale = TRUE)

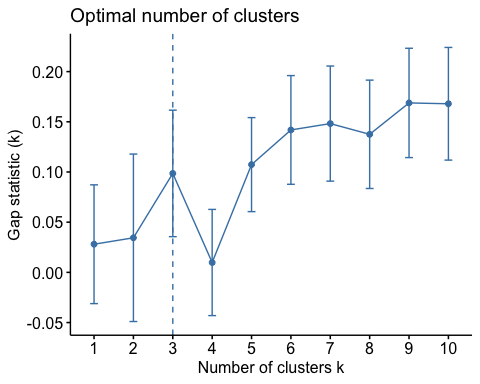
The code chunk above calls the appropriate dataset, removes any potential missing observations, checks if scaling is necessary and scales the data.

### Problem 4

# Create Dissimilarity matrix  
diss.matrix = dist(us\_arrest, method = "euclidean")

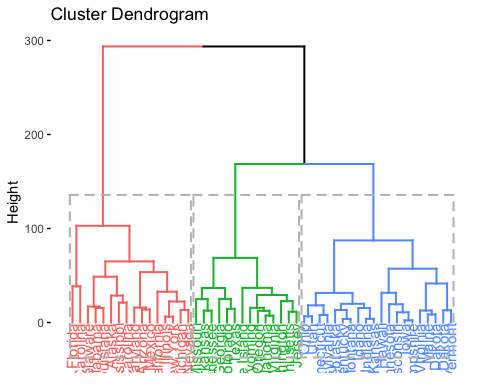
#### Part 1: Complete Linkage

#Identifying the optimal number of clusters given complete linkage method  
set.seed(100)  
gap\_stat\_c = clusGap(us\_arrest, FUN = hcut, hc\_method = "complete", K.max = 10, B = 50)  
fviz\_gap\_stat(gap\_stat\_c)

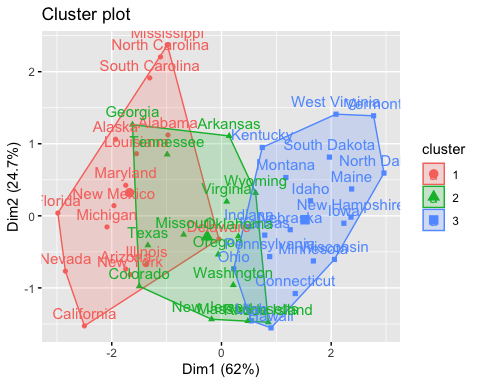


After running the above hierarchical clustering model using the “complete” linkage method, the optimal number of clusters per the gap statistic is *3*.

#Characterizing the clusters  
clusters.c = hcut(us\_arrest, k = 3, hc\_func = "hclust", hc\_method = "complete", hc\_metric = "euclidian")  
  
clusters.c$size  
fviz\_dend(clusters.c, rect = TRUE)



fviz\_cluster(clusters.c)



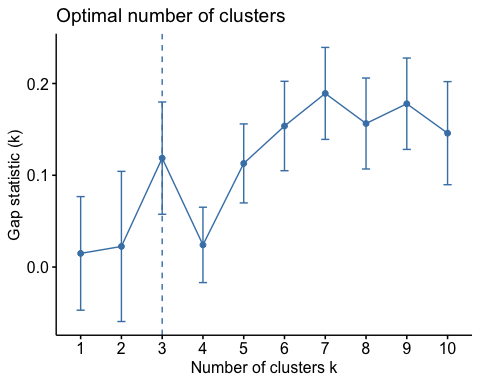
input.feature.vals = cbind(us\_arrest, cluster = clusters.c$cluster)  
input.feature.vals %>%  
 group\_by(cluster) %>%  
 summarise\_all(mean) %>%   
 knitr::kable()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cluster | Murder | Assault | UrbanPop | Rape |
| 1 | 11.812500 | 272.5625 | 68.31250 | 28.37500 |
| 2 | 8.214286 | 173.2857 | 70.64286 | 22.84286 |
| 3 | 4.270000 | 87.5500 | 59.75000 | 14.39000 |

Within cluster 1, there are 16 states included; cluster 2 includes 14 states, and cluster 3 includes 20 states. The pattern that emerges from clusters seems to be that cluster 1 contains states with the greatest number of arrests for murder, assault and rape and also has the highest urban population, on average. Cluster 2 has the second-highest number of arrests for murder, assault and rape and the second-greatest urban population, on average. Lastly, cluster 3 has the least number of arrests for murder, assault and rape and the lowest urban population, on average.

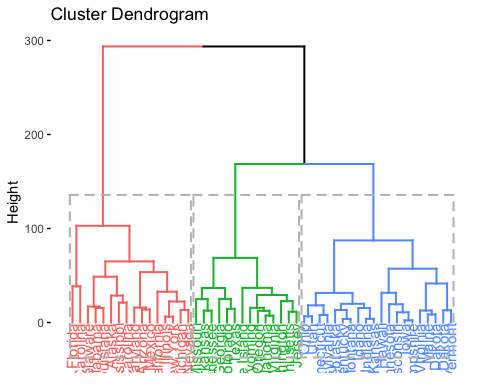
#### Part 2: Average Linkage

#Identifying the optimal number of clusters given average linkage method  
gap\_stat = clusGap(us\_arrest, FUN = hcut, hc\_method = "average", K.max = 10, B = 50)  
fviz\_gap\_stat(gap\_stat)

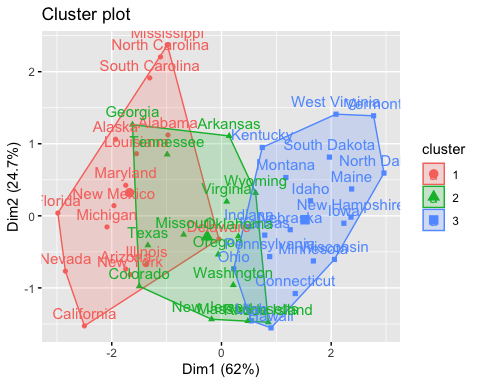


After running the above hierarchical clustering model using the “average” linkage method, the optimal number of clusters per the gap statistic is *3*.

#Characterizing the clusters  
clusters.a = hcut(us\_arrest, k = 3, hc\_func = "hclust", hc\_method = "complete", hc\_metric = "euclidian")  
  
clusters.a$size  
fviz\_dend(clusters.a, rect = TRUE)



fviz\_cluster(clusters.a)



input.feature.vals = cbind(us\_arrest, cluster = clusters.a$cluster)  
input.feature.vals %>%  
 group\_by(cluster) %>%  
 summarise\_all(mean) %>%   
 knitr::kable()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cluster | Murder | Assault | UrbanPop | Rape |
| 1 | 11.812500 | 272.5625 | 68.31250 | 28.37500 |
| 2 | 8.214286 | 173.2857 | 70.64286 | 22.84286 |
| 3 | 4.270000 | 87.5500 | 59.75000 | 14.39000 |

Within cluster 1, there are 20 states included; cluster 2 includes 14 states, and cluster 3 includes 16 states. The mean values given per feature among clusters are the same for the average and complete linkage methods. Given these results, there are no differences between using the complete or average linkage method, as both point towards using 3 clusters.

### Problem 5

Using the clusters as exposures, we could assess whether different arrest rates are resulting in poorer health outcomes. I would potentially want to investigate attitudes towards law enforcement and arrest rates by cluster, to determine whether arrests are increasing or decreasing positive attitudes towards law enforcement. In terms of scientific issues, we should ensure to weight by population in order to drawing conclusions that are simply the result of a higher urban population. We should also consider whether this data includes convicted individuals or whether it is based on charges. Particularly when considering the under-reporting of sexual assault and rape instances, it’s important to understand whether this data captures charges or convicted cases. Additionally, we should break down what “assault” refers to, as different forms of assaults have different implications in terms of preventive measures; interventions for the prevention of domestic violence will look different than interventions for preventing robberies.