ei2262\_assignment4

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

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.1 ✔ purrr 0.3.5   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.0 ✔ stringr 1.4.1   
## ✔ readr 2.1.2 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(stats)  
library(factoextra)

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

library(cluster)  
library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

set.seed(123)

### Part 1: Implementing a Simple Prediction Pipeline

#### Loading in class4\_p1.csv Dataset and Performing Basic Data Cleaning

class4\_p1 = read\_csv("./class4\_p1.csv")

## New names:  
## Rows: 3811 Columns: 17  
## ── Column specification  
## ──────────────────────────────────────────────────────── Delimiter: "," dbl  
## (17): ...1, chronic1, chronic3, chronic4, bmi, tobacco1, alcohol1, gpaq8...  
## ℹ Use `spec()` to retrieve the full column specification for this data. ℹ  
## Specify the column types or set `show\_col\_types = FALSE` to quiet this message.  
## • `` -> `...1`

#Checking the variable types  
str(class4\_p1)

## spc\_tbl\_ [3,811 × 17] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ ...1 : num [1:3811] 1 2 3 4 5 6 7 8 9 10 ...  
## $ chronic1 : num [1:3811] 2 2 1 2 2 1 1 1 1 2 ...  
## $ chronic3 : num [1:3811] 2 2 1 2 2 1 1 1 1 2 ...  
## $ chronic4 : num [1:3811] 2 2 2 1 2 2 2 2 2 2 ...  
## $ bmi : num [1:3811] 26.8 28.2 39.2 42.5 22.3 ...  
## $ tobacco1 : num [1:3811] 3 3 1 3 3 3 3 3 3 2 ...  
## $ alcohol1 : num [1:3811] 3 3 2 3 NA 2 2 3 2 2 ...  
## $ gpaq8totmin : num [1:3811] 0 120 60 180 0 0 0 0 15 0 ...  
## $ gpaq11days : num [1:3811] 5 5 3 5 7 7 7 5 5 7 ...  
## $ habits5 : num [1:3811] 3 1 1 2 2 3 2 3 4 2 ...  
## $ habits7 : num [1:3811] NA NA NA NA NA NA NA NA NA NA ...  
## $ agegroup : num [1:3811] 1 3 3 2 2 3 3 4 3 2 ...  
## $ dem3 : num [1:3811] 2 2 1 2 2 1 1 1 2 1 ...  
## $ dem4 : num [1:3811] 1 2 1 1 2 2 1 1 1 1 ...  
## $ dem8 : num [1:3811] 2 2 1 2 2 1 2 2 1 1 ...  
## $ povertygroup: num [1:3811] 1 3 6 2 6 3 4 6 3 1 ...  
## $ healthydays : num [1:3811] 24 30 NA 23 30 30 30 10 0 30 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. ...1 = col\_double(),  
## .. chronic1 = col\_double(),  
## .. chronic3 = col\_double(),  
## .. chronic4 = col\_double(),  
## .. bmi = col\_double(),  
## .. tobacco1 = col\_double(),  
## .. alcohol1 = col\_double(),  
## .. gpaq8totmin = col\_double(),  
## .. gpaq11days = col\_double(),  
## .. habits5 = col\_double(),  
## .. habits7 = col\_double(),  
## .. agegroup = col\_double(),  
## .. dem3 = col\_double(),  
## .. dem4 = col\_double(),  
## .. dem8 = col\_double(),  
## .. povertygroup = col\_double(),  
## .. healthydays = col\_double()  
## .. )  
## - attr(\*, "problems")=<externalptr>

#Converting categorical variables into factors  
class4\_p1 =  
 class4\_p1 %>%   
 mutate(  
 chronic1 = as.factor(chronic1),  
 chronic3 = as.factor(chronic3),  
 chronic4 = as.factor(chronic4),  
 tobacco1 = as.factor(tobacco1),  
 alcohol1 = as.factor(alcohol1),  
 habits5 = as.factor(habits5),  
 habits7 = as.factor(habits7),  
 agegroup = as.factor(agegroup),  
 dem3 = as.factor(dem3),  
 dem4 = as.factor(dem4),  
 dem8 = as.factor(dem8),  
 povertygroup = as.factor(povertygroup))  
  
#Checking to make sure variables were properly converted to factors  
str(class4\_p1)

## tibble [3,811 × 17] (S3: tbl\_df/tbl/data.frame)  
## $ ...1 : num [1:3811] 1 2 3 4 5 6 7 8 9 10 ...  
## $ chronic1 : Factor w/ 2 levels "1","2": 2 2 1 2 2 1 1 1 1 2 ...  
## $ chronic3 : Factor w/ 2 levels "1","2": 2 2 1 2 2 1 1 1 1 2 ...  
## $ chronic4 : Factor w/ 2 levels "1","2": 2 2 2 1 2 2 2 2 2 2 ...  
## $ bmi : num [1:3811] 26.8 28.2 39.2 42.5 22.3 ...  
## $ tobacco1 : Factor w/ 3 levels "1","2","3": 3 3 1 3 3 3 3 3 3 2 ...  
## $ alcohol1 : Factor w/ 3 levels "1","2","3": 3 3 2 3 NA 2 2 3 2 2 ...  
## $ gpaq8totmin : num [1:3811] 0 120 60 180 0 0 0 0 15 0 ...  
## $ gpaq11days : num [1:3811] 5 5 3 5 7 7 7 5 5 7 ...  
## $ habits5 : Factor w/ 4 levels "1","2","3","4": 3 1 1 2 2 3 2 3 4 2 ...  
## $ habits7 : Factor w/ 5 levels "1","2","3","4",..: NA NA NA NA NA NA NA NA NA NA ...  
## $ agegroup : Factor w/ 4 levels "1","2","3","4": 1 3 3 2 2 3 3 4 3 2 ...  
## $ dem3 : Factor w/ 2 levels "1","2": 2 2 1 2 2 1 1 1 2 1 ...  
## $ dem4 : Factor w/ 2 levels "1","2": 1 2 1 1 2 2 1 1 1 1 ...  
## $ dem8 : Factor w/ 2 levels "1","2": 2 2 1 2 2 1 2 2 1 1 ...  
## $ povertygroup: Factor w/ 6 levels "1","2","3","4",..: 1 3 6 2 6 3 4 6 3 1 ...  
## $ healthydays : num [1:3811] 24 30 NA 23 30 30 30 10 0 30 ...

#Omit missing data  
class4.nomiss<-na.omit(class4\_p1)

For the two prediction models, I have chosen the following features to include in the linear regression model:

Model 1 = Age (agegroup), How physically active are you? (habits5), In general, how healthy is your overall diet? (habits7), BMI (bmi), Minutes of total physical activity on home chores on an average day (gpaq8totmin), How many days did you walk to get to and from places in the last 7 days? (gpaq11days)

Model 2 = Age (Agegroup), BMI (bmi), Do you usually smoke 3 or more cigarettes on most days, some days, or never? (tobacco1), Do you usually have more than 2 drinks of alcohol on most days, some days, or never? (alcohol1), Do you currently have hypertension? (chronic1), Do you currently have diabetes diagnosed by a medical professional? (chronic3), In the last 12 months, have you had an episode of asthma or an asthma attack? (chronic4)

#### Setting up Data for Analysis

The numeric variables will be scaled and centered.

class4.numeric<- class4.nomiss %>% dplyr::select(where(is.numeric))  
  
set.up.preprocess<-preProcess(class4.numeric, method=c("center", "scale"))  
transformed.vals<-predict(set.up.preprocess, class4.numeric)

#### Partitioning Outcome Variable (Healthydays) into Training and Testing (70/30 Split)

train.index<-createDataPartition(class4.nomiss$healthydays, p=0.7, list=FALSE)  
  
#Training set (70%)  
class4.train<-class4.nomiss[train.index,]  
  
#Testing set (30%)  
class4.test<-class4.nomiss[-train.index,]

#### Model Training and Testing

Prediction Model 1: Agegroup, Habits5, Habits7, BMI, gpaq8totmin, gpaq11days This model includes a subset of features that focus on age, BMI, and the amount and type of physical activity an individual reported.

model\_1 = lm(healthydays ~ agegroup + habits5 + habits7 + bmi + gpaq8totmin + gpaq11days, data = class4.train)  
  
fitted\_results1 = predict(model\_1, class4.train, type = 'response')  
  
fitted\_results2 = postResample(fitted\_results1, class4.train$healthydays)  
fitted\_results2

## RMSE Rsquared MAE   
## 7.2844175 0.1157276 4.6376236

Prediction Model 2: Agegroup, BMI, tobacco1, alcohol1, chronic1, chronic3, chronic4 This model includes a subset of features that focus on age, BMI, smoking and drinking patterns, and underlying chronic diagnoses (diabetes, hypertension, asthma)

model\_2 = lm(healthydays ~ agegroup + bmi + tobacco1 + alcohol1 + chronic1 + chronic3 + chronic4, data = class4.train)   
  
fitted\_results3 = predict(model\_2, class4.train, type = 'response')  
  
fitted\_results4 = postResample(fitted\_results3, class4.train$healthydays)  
fitted\_results4

## RMSE Rsquared MAE   
## 7.38843925 0.09029239 4.65325374

After applying both models within the test data, it was determined that Model 1 was the preferred prediction model. Model 1 had the lower root mean square error (RMSE) value, 7.2186 vs. 7.23934 (Model 2). Model 1 also had the higher R^2 value, 0.09527 vs. 0.09006 (Model 2). Since Model 1 has the lower RMSE as well as the higher R^2 value, it is the preferred prediction model over Model 2.

### Q3: Describe one setting (in 1-2 sentences) where the implementation of your final model would be useful.

The implementation of my final model would be useful when the researchers are exploring the effects of physical activity, age, and BMI on individuals reporting having good physical health. This would be useful information in creating a program that focuses on promoting the mental and physical benefits of moving our body, whether it is through everyday activities like chores or physical activity like weight training.

### Part 2: Conducting an Unsupervised Analysis

#### Loading in US Arrests Dataset and Performing Basic Data Cleaning

set.seed(123)  
us\_arrests <- USArrests  
  
us\_arrests <-na.omit(us\_arrests)  
  
#Checking if we need to scale and center  
colMeans(us\_arrests, na.rm=TRUE)

## Murder Assault UrbanPop Rape   
## 7.788 170.760 65.540 21.232

apply(us\_arrests, 2, sd, na.rm=TRUE)

## Murder Assault UrbanPop Rape   
## 4.355510 83.337661 14.474763 9.366385

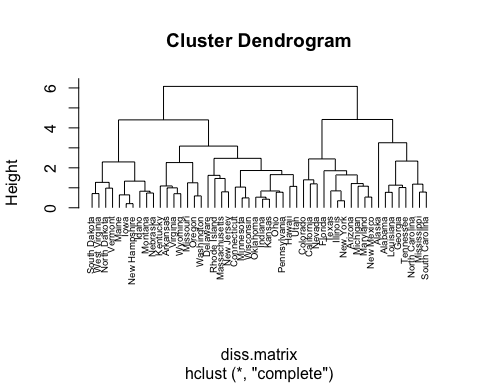
arrests\_scaled<-scale(us\_arrests)

Scaled and centered the dataset because there are differences in mean and standard deviation. To avoid any dependency on an arbitrury variable, I used scale to standardize the dataset.

#### Agglomerative Algorithm for Hierarchical Clustering

Hierarchical Clustering will be conducted using complete Linkage. Euclidian distance measure will be used to construct the dissimilar matrix

#Creating Dissimilar Matrix  
diss.matrix <- dist(arrests\_scaled, method = "euclidean")  
  
clusters.arrests<- hclust(diss.matrix, method = "complete" )  
  
#Plotting Cluster Dendrogram to see Clusters  
plot(clusters.arrests, cex = 0.6, hang = -1)



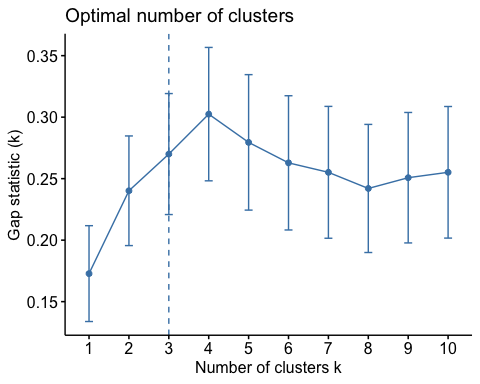
#### Q4a: Determine the Optimal Number of Clusters Using a Clear, Data-Driven Strategy.

The optimal number of clusters will be determined using Gap Statistics. I will plot and print the optimal number of clusters to identify the value of k (# of clusters) that has the highest gap statistic.

gap\_stat <- clusGap(arrests\_scaled, FUN = hcut, nstart = 25, K.max = 10, B = 50)  
print(gap\_stat, method="firstmax")

## Clustering Gap statistic ["clusGap"] from call:  
## clusGap(x = arrests\_scaled, FUNcluster = hcut, K.max = 10, B = 50, nstart = 25)  
## B=50 simulated reference sets, k = 1..10; spaceH0="scaledPCA"  
## --> Number of clusters (method 'firstmax'): 4  
## logW E.logW gap SE.sim  
## [1,] 3.458369 3.631167 0.1727975 0.03896440  
## [2,] 3.142869 3.383016 0.2401470 0.04457094  
## [3,] 2.978518 3.248506 0.2699880 0.04918469  
## [4,] 2.833430 3.135886 0.3024565 0.05422817  
## [5,] 2.759739 3.039231 0.2794925 0.05507099  
## [6,] 2.689377 2.952214 0.2628371 0.05456848  
## [7,] 2.616505 2.871649 0.2551443 0.05360321  
## [8,] 2.556388 2.798408 0.2420191 0.05207751  
## [9,] 2.478796 2.729565 0.2507688 0.05303471  
## [10,] 2.408612 2.663773 0.2551613 0.05349496

fviz\_gap\_stat(gap\_stat)

 Using gap statistics, it was determined that 4 was the optimal number of clusters. At [4,], the gap statistic is the highest value (0.3025). The graph also shows that the highest gap statistic at k = 4.

#### Q4b: Describe the composition of each cluster in terms of the original input features.

I will use the number of clusters from the gap statistic to obtain cluster assignment for each observation.

#Identifying the number of observations in each cluster  
clusters.arrests4<-cutree(clusters.arrests, k=4)  
table(clusters.arrests4)

## clusters.arrests4  
## 1 2 3 4   
## 8 11 21 10

#Attaching a label of each state to identify the cluster assignment for each observation.  
arrests\_labeled <- cbind(USArrests, cluster = clusters.arrests4)  
  
arrests\_labeled %>%   
 group\_by(cluster) %>%  
 summarise\_all(mean) %>%   
 knitr::kable(caption = "Composition of Each Cluster")

Composition of Each Cluster

| cluster | Murder | Assault | UrbanPop | Rape |
| --- | --- | --- | --- | --- |
| 1 | 14.087500 | 252.7500 | 53.50000 | 24.53750 |
| 2 | 11.054545 | 264.0909 | 79.09091 | 32.61818 |
| 3 | 5.871429 | 134.4762 | 70.76190 | 18.58095 |
| 4 | 3.180000 | 78.7000 | 49.30000 | 11.63000 |

There are a total of 4 clusters. There are 8 states in Cluster 1, 11 states in Cluster 2, 21 states in Cluster 3, and 10 states in Cluster 4.

Cluster 1: Alabama, Alaska, Georgia, Louisiana, Mississippi, North Carolina, South Carolina, and Tennessee

arrests\_labeled %>%   
 select(cluster) %>%   
 filter(cluster == 1) %>%   
 knitr::kable(caption = "Cluster 1")

Cluster 1

|  | cluster |
| --- | --- |
| Alabama | 1 |
| Alaska | 1 |
| Georgia | 1 |
| Louisiana | 1 |
| Mississippi | 1 |
| North Carolina | 1 |
| South Carolina | 1 |
| Tennessee | 1 |

Cluster 2: Arizona, California, Colorado, Florida, Illinois, Maryland, Michigan, Nevada, New Mexico, New York, and Texas

arrests\_labeled %>%   
 select(cluster) %>%   
 filter(cluster == 2) %>%   
 knitr::kable(caption = "Cluster 2")

Cluster 2

|  | cluster |
| --- | --- |
| Arizona | 2 |
| California | 2 |
| Colorado | 2 |
| Florida | 2 |
| Illinois | 2 |
| Maryland | 2 |
| Michigan | 2 |
| Nevada | 2 |
| New Mexico | 2 |
| New York | 2 |
| Texas | 2 |

Cluster 3: Arkansas, Connecticut, Delaware, Hawaii, Indiana, Kansas, Kentucky, Massachusetts, Minnesota, Missouri, New Jersey, Ohio, Oklahoma, Oregon, Pennsylvania, Rhode Island, Utah, Virginia, Washington, Wisconsin, and Wyoming

arrests\_labeled %>%   
 select(cluster) %>%   
 filter(cluster == 3) %>%   
 knitr::kable(caption = "Cluster 3")

Cluster 3

|  | cluster |
| --- | --- |
| Arkansas | 3 |
| Connecticut | 3 |
| Delaware | 3 |
| Hawaii | 3 |
| Indiana | 3 |
| Kansas | 3 |
| Kentucky | 3 |
| Massachusetts | 3 |
| Minnesota | 3 |
| Missouri | 3 |
| New Jersey | 3 |
| Ohio | 3 |
| Oklahoma | 3 |
| Oregon | 3 |
| Pennsylvania | 3 |
| Rhode Island | 3 |
| Utah | 3 |
| Virginia | 3 |
| Washington | 3 |
| Wisconsin | 3 |
| Wyoming | 3 |

Cluster 4: Idaho, Iowa, Maine, Montana, Nebraska, New Hampshire, North Dakota, South Dakota, Vermont, and West Virginia

arrests\_labeled %>%   
 select(cluster) %>%   
 filter(cluster == 4) %>%   
 knitr::kable(caption = "Cluster 4")

Cluster 4

|  | cluster |
| --- | --- |
| Idaho | 4 |
| Iowa | 4 |
| Maine | 4 |
| Montana | 4 |
| Nebraska | 4 |
| New Hampshire | 4 |
| North Dakota | 4 |
| South Dakota | 4 |
| Vermont | 4 |
| West Virginia | 4 |

#### Q5: Pretend that the data are from 2020 and not 1973. Describe one research question that can be addressed using the newly identified clusters. Briefly comment on any scientific or ethical considerations one should review before using these clusters for your specific question. NOTE: The clusters can be used as an exposure, an outcome, or a covariate.

If this data is from 2020, one research question that can be addressed using the newly identified clusters is, “Do states with similar violence profiles predict higher incidence of mental health disorders?”

Some considerations one should review before using these clusters for this specific question include the effects socioeconomic status of the residents in each cluster has on the statistics. USArrests dataset are statistics on the arrests per 100.000 residents for assault, murder, and rape in each of the 50 US states in 1973. If this data was collected in 2020, we must be cautious of the socioeconomic status of states and regions that have higher rates of arrests. We must also take into consideration racial disparities in the judicial system.