Assignment-4

2024-02-13

library(readr)  
library(dplyr)  
library(Amelia)  
library(factoextra)  
library(cluster)  
library(caret)

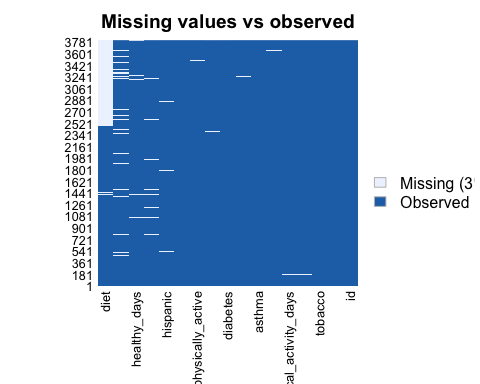
### Part I: Implementing a Simple Prediction Pipeline

1. Perform basic data cleaning.

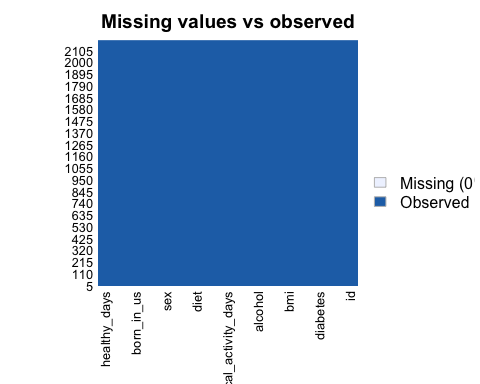
# path where datasets are stored as working directory  
setwd("/Users/apple/Desktop/8451 Machine Learning for Epi/Assignment-4")  
  
# read dataset  
data <- read\_csv("./class4\_p1.csv")  
  
# rename the variable names to make them more easy to understand  
var.names <- c("id","hypertension","diabetes","asthma","bmi","tobacco","alcohol","physical\_activity\_total\_minutes","physical\_activity\_days","physically\_active","diet","agegroup","sex","hispanic","born\_in\_us","poverty\_group","healthy\_days")  
  
colnames(data) <- var.names  
  
# check which features are continuous and which are categorical  
data |> str()

## spc\_tbl\_ [3,811 × 17] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ id : num [1:3811] 1 2 3 4 5 6 7 8 9 10 ...  
## $ hypertension : num [1:3811] 2 2 1 2 2 1 1 1 1 2 ...  
## $ diabetes : num [1:3811] 2 2 1 2 2 1 1 1 1 2 ...  
## $ asthma : 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 ...  
## $ tobacco : num [1:3811] 3 3 1 3 3 3 3 3 3 2 ...  
## $ alcohol : num [1:3811] 3 3 2 3 NA 2 2 3 2 2 ...  
## $ physical\_activity\_total\_minutes: num [1:3811] 0 120 60 180 0 0 0 0 15 0 ...  
## $ physical\_activity\_days : num [1:3811] 5 5 3 5 7 7 7 5 5 7 ...  
## $ physically\_active : num [1:3811] 3 1 1 2 2 3 2 3 4 2 ...  
## $ diet : 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 ...  
## $ sex : num [1:3811] 2 2 1 2 2 1 1 1 2 1 ...  
## $ hispanic : num [1:3811] 1 2 1 1 2 2 1 1 1 1 ...  
## $ born\_in\_us : num [1:3811] 2 2 1 2 2 1 2 2 1 1 ...  
## $ poverty\_group : num [1:3811] 1 3 6 2 6 3 4 6 3 1 ...  
## $ healthy\_days : 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>

# The results from str() show all the features are numeric. However, all features except for id, bmi, physical\_activity\_total\_minutes, physical\_activity\_days, and healthy\_days need to be categorical.   
  
# convert categorical variables that have been read-in as continous variables  
data <- data |> mutate\_at(vars(id,hypertension,diabetes,asthma,tobacco,alcohol,  
 physically\_active,diet,agegroup,sex,hispanic,born\_in\_us,  
 poverty\_group), as.factor)  
  
# check missing data  
data |> missmap(main = "Missing values vs observed")



# The missing map shows only 3% of the data are missing. However, we may still want to omit missing value.   
  
# omit missing value  
data <- data |> na.omit()  
  
# check missing data age  
data |> missmap(main = "Missing values vs observed")



# Now, there is no missing value.  
  
# check whether there is only one id per person  
data <- data |> distinct(id, .keep\_all = TRUE)   
  
# No change of the observation's number indicates there was no duplicate in the dataset.  
  
# remove the id variable  
data$id <- NULL

1. Partition data into training and testing (use a 70/30 split)

# set seed  
set.seed(123)  
  
# split data into training and testing  
train.index = createDataPartition(data$healthy\_days, p = 0.7, list = FALSE)  
  
train = data[train.index, ]  
test = data[-train.index, ]

1. Fit two prediction models using different subsets of the features in the training data.

**Discussion:**

I will use two linear regression models to do the machine learning task. The dependent variable in both models is the number of healthy days. Model 1 contains two independent variables which are whether or not have hypertension and number of physical active days (during the last 7 days, on how many days did you walk to get to and from places). Model 2 contains all the features available in the dataset.

* Model 1 : healthy\_days ~ hypertension + physical\_activity\_days
* Model 2 : healthy\_days ~ hypertension + diabetes + asthma + tobacco + alcohol + physically\_active + diet + agegroup + sex + hispanic + born\_in\_us + poverty\_group

# Avoid overfitting by repeated cross-validation  
control = trainControl(method="repeatedcv", number=10, repeats=10, summaryFunction=defaultSummary)  
  
# fit prediction models  
model1 = train(healthy\_days ~ hypertension + physical\_activity\_days, data = train, method = "lm", trControl = control)  
model2 = train(healthy\_days ~ hypertension + diabetes + asthma + tobacco +   
 alcohol + physically\_active + diet + agegroup + sex +   
 hispanic + born\_in\_us + poverty\_group, data = train, method = "lm", trControl = control)

1. Apply both models within the test data and determine which model is the preferred prediction model using the appropriate evaluation metric(s).

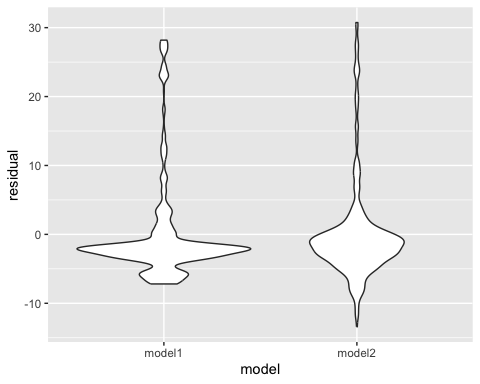
# make predictions within test dataset  
prediction1 = predict(model1, test)  
prediction2 = predict(model2, test)  
  
# evaluate the models using mean squared error  
mse1 = mean((prediction1 - test$healthy\_days)\*\*2/nrow(test)) |> print()

## [1] 0.08535169

mse2 = mean((prediction2 - test$healthy\_days)\*\*2/nrow(test)) |> print()

## [1] 0.07830613

model1\_result = bind\_rows(train = prediction1, test = test$healthy\_days, .id = "split")  
model2\_result = bind\_rows(train = prediction2, test = test$healthy\_days, .id = "split")  
result = bind\_rows(model1 = model1\_result, model2 = model2\_result, .id = "model")  
  
# calculate the residual  
result = result |>   
 mutate(residual = train - test)  
  
# plot the residual  
result |>   
 ggplot(aes(x = model, y = residual)) + geom\_violin()



**Discussion:** The mean squared error (MSE) for Model 1 (0.085) is slightly higher than that of Model 2 (0.078). Additionally, the residual plot indicates that both models have residuals centered around zero, which is a desirable outcome. Moreover, the spread of the residuals, as depicted by the range from top to bottom of the violin plot, appears similar for both models, suggesting comparable variance in their predictions. However, upon closer examination, Model 2 exhibits a slightly tighter distribution of residuals, implying more consistent predictions. Therefore, based on the MSE evaluation, Model 2 is considered the preferred prediction model.

1. Describe one setting where the implementation of your final model would be useful.

**Discussion:** The Model 2 is capable of predicting the number of days an individual reports having good physical health in a month. This prediction is derived from a comprehensive evaluation of medical conditions, lifestyle habits, and sociodemographic factors. The insights gleaned from this model can inform targeted health interventions and resource allocation strategies aimed at enhancing the overall well-being of the population in New York City.

### Part II: Conducting an Unsupervised Analysis

Using the dataset from the Group assignment Part 3 (USArrests), identify clusters using hierarchical analysis.

1. Conduct a hierarchical clustering analysis. Use a Euclidian distance measure to construct your dissimilarity matrix. Use complete linkage.

# import dataset  
data("USArrests")  
  
# strip off the outcome and id variable  
USArrests <- USArrests |> select("Murder", "Assault", "UrbanPop", "Rape")  
  
# determine if scale is necessary  
USArrests |> summarise\_all(mean, na.rm = TRUE) |>  
 print()

## Murder Assault UrbanPop Rape  
## 1 7.788 170.76 65.54 21.232

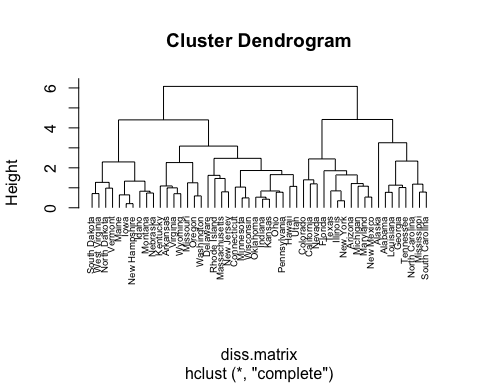
USArrests |> summarise\_all(sd, na.rm = TRUE) |>  
 print()

## Murder Assault UrbanPop Rape  
## 1 4.35551 83.33766 14.47476 9.366385

# Since the mean and standard deviation's magnitude are different for each variable. So, we decide to center and scale.  
  
# set seed  
set.seed(123)  
  
# center and scale  
set.up.preprocess = preProcess(USArrests, method = c("center", "scale"))  
  
# output pre-processed values  
transformed.vals = predict(set.up.preprocess, USArrests)  
  
# compare the sds used to scale to the sds above to ensure they are close   
USArrests$scale

## NULL

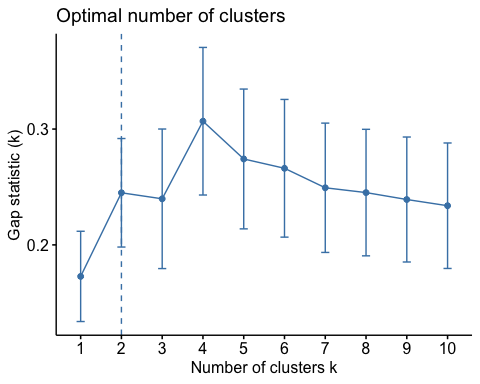
# The sds used to scale to the sds above are close.  
  
# conduct clustering analysis using hierarchical clustering by creating dissimilarity matrix first  
diss.matrix <- dist(transformed.vals, method = "euclidean")  
  
# hierarchical clustering using complete linkage  
clusters.hcut <- hclust(diss.matrix, method = "complete")  
  
# plot the obtained dendrogram  
plot(clusters.hcut, cex = 0.6, hang = -1)



1. Determine the optimal number of clusters using a clear, data-driven strategy.

* We could determine the optimal number of cluster by looking at the Gap Statistic Graph. Since when number of clusters k = 4, it gives us the highest Gap Statistic. Therefore, the optimal number of clusters is 4.

# use complete method  
gap\_stat = clusGap(transformed.vals, FUN = hcut, hc\_method="complete", K.max = 10, B = 50)  
fviz\_gap\_stat(gap\_stat)



# use number of clusters from gap statistic to obtain cluster assignment for each observation  
clusters.h.4 <- cutree(clusters.hcut, k=4)  
table(clusters.h.4)

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

# set k to be 4  
clusters.hcut=hcut(transformed.vals, k=4, hc\_func="hclust", hc\_method="complete", hc\_metric="euclidian")

1. Describe the composition of each cluster in terms of the original input features

**Discussion:**

* Cluster 1: This cluster is characterized by above-average values in all categories except UrbanPop, which is close to the average. It suggests states with higher crime rates but moderately smaller urban populations.
* Cluster 2: States in this cluster exhibit all features above the average, especially assault and rape rates. This cluster likely represents states with high crime rates and large urban populations.
* Cluster 3: Negative z-scores for murder, assault, and rape rate indicate that these areas have lower than average rates of these three types of crimes. The UrbanPop is slightly above average, suggesting moderately sized urban populations but low crime rates in these areas.
* Cluster 4: This cluster is characterized by below-average values for all features (Murder, Assault, UrbanPop, Rape), as indicated by the negative z-scores. States falling into this cluster tend to have lower than average crime rates and small urban population sizes.

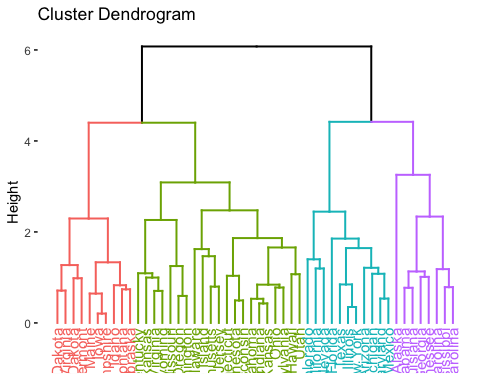
# show the mean value of features within each cluster  
# bind the cluster value to original data  
input.feature.vals=cbind(transformed.vals,cluster=clusters.hcut$cluster)  
  
# summarize the mean value of features within each cluster  
input.feature.vals |> group\_by(cluster) |>   
 summarise\_all(mean)

## # A tibble: 4 × 5  
## cluster Murder Assault UrbanPop Rape  
## <int> <dbl> <dbl> <dbl> <dbl>  
## 1 1 1.45 0.984 -0.832 0.353  
## 2 2 0.750 1.12 0.936 1.22   
## 3 3 -0.440 -0.435 0.361 -0.283  
## 4 4 -1.06 -1.10 -1.12 -1.03

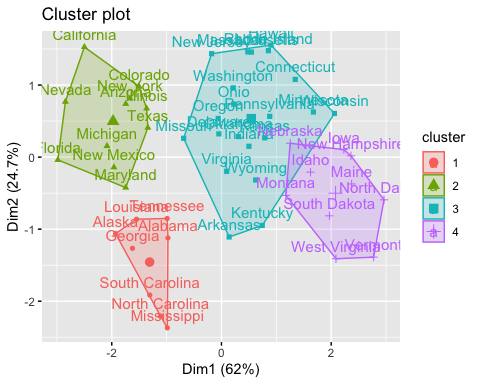
# check size of the clusters  
clusters.hcut$size

## [1] 8 11 21 10

# check cluster dendrogram  
fviz\_dend(clusters.hcut)



# check cluster plot  
fviz\_cluster(clusters.hcut)



1. Pretend that the data are from 2022 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.

**Discussion:**

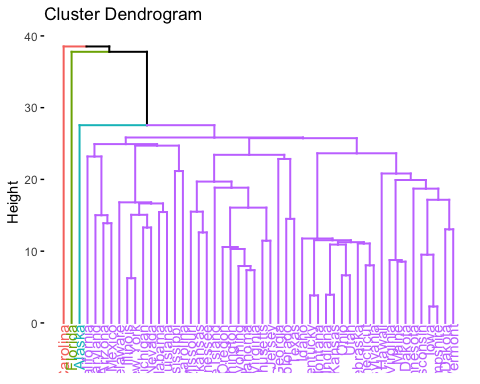
* We can investigate if the relationship between the outcome urban population size and indicators for crime such as murder, assault, and rape rate varies by clusters. If the data are from 2022, there might be more features available such as sociodemographic data as well. So, in terms of scientific considerations, we need to consider which feature we should select and how many clusters there need to have. In terms of ethical considerations, we should check whether the data is representative for all sociodemographic groups in the society to avoid a biased dataset where certain groups are neglected.

1. Optional Repeat analysis with a different linkage method (e.g. single or average). Do the clusters change?

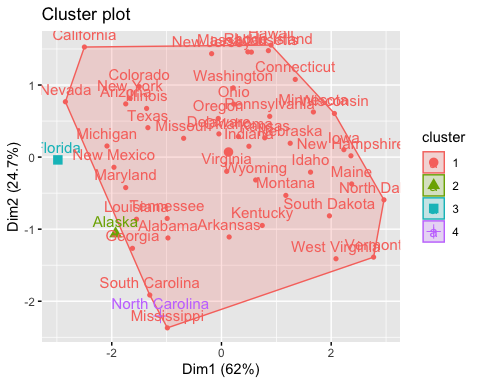
# change the linkage method to single   
clusters.hcut.4 <-hcut(USArrests, k = 4, hc\_func = "hclust", hc\_method = "single", hc\_metric = "euclidian")  
  
# check size of the clusters  
clusters.hcut.4$size

## [1] 47 1 1 1

# check cluster dendrogram  
fviz\_dend(clusters.hcut.4)



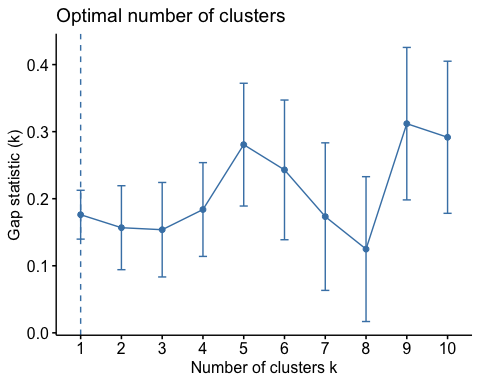
# check cluster plot  
fviz\_cluster(clusters.hcut.4)



# bind the cluster value to original data  
input.feature.vals <-cbind(transformed.vals,cluster = clusters.hcut.4$cluster)  
  
# summarize the mean value of features within each cluster  
input.feature.vals |> group\_by(cluster) |>   
 summarise\_all(mean)

## # A tibble: 4 × 5  
## cluster Murder Assault UrbanPop Rape  
## <int> <dbl> <dbl> <dbl> <dbl>  
## 1 1 -0.0735 -0.108 0.0347 -0.0654  
## 2 2 0.508 1.11 -1.21 2.48   
## 3 3 1.75 1.97 0.999 1.14   
## 4 4 1.20 1.99 -1.42 -0.548

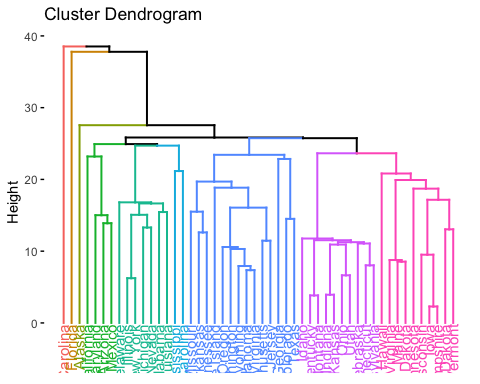
# Since 4 clusters may not delineate distinct patterns, we want to search for the best number of clusters.  
gap\_stat <- clusGap(transformed.vals, FUN = hcut, hc\_method="single", K.max = 10, B = 50)  
fviz\_gap\_stat(gap\_stat)



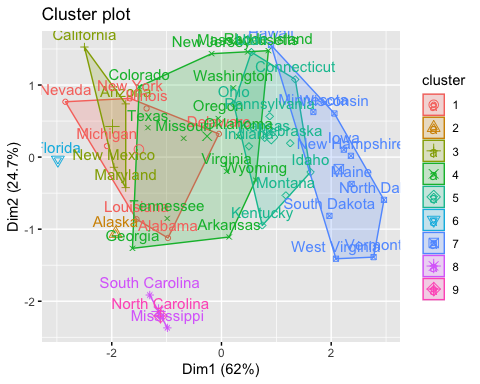
# 9 has the highest Gap Statistics  
  
# repeat the previous code for 9 clusters  
clusters.hcut.9 <-hcut(USArrests, k = 9, hc\_func = "hclust", hc\_method = "single", hc\_metric = "euclidian")  
  
# check size of the clusters  
clusters.hcut.9$size

## [1] 7 1 4 14 10 1 10 2 1

# check cluster dendrogram  
fviz\_dend(clusters.hcut.9)



# check cluster plot  
fviz\_cluster(clusters.hcut.9)



# bind the cluster value to original data  
input.feature.vals <-cbind(transformed.vals,cluster = clusters.hcut.9$cluster)  
  
# summarize the mean value of features within each cluster  
input.feature.vals |> group\_by(cluster) |>   
 summarise\_all(mean)

## # A tibble: 9 × 5  
## cluster Murder Assault UrbanPop Rape  
## <int> <dbl> <dbl> <dbl> <dbl>  
## 1 1 0.846 0.922 0.604 0.637  
## 2 2 0.508 1.11 -1.21 2.48   
## 3 3 0.496 1.42 0.792 1.24   
## 4 4 0.0979 0.0303 0.353 0.172  
## 5 5 -0.505 -0.700 0.00415 -0.423  
## 6 6 1.75 1.97 0.999 1.14   
## 7 7 -1.11 -1.30 -0.804 -1.04   
## 8 8 1.71 1.18 -1.35 -0.153  
## 9 9 1.20 1.99 -1.42 -0.548

**Discussion:**

The clusters change after using a different linkage method, specifically the single linkage method. With the complete linkage method, specifying a desire for 4 clusters results in clusters of relatively similar size, comprising 8, 11, 21, and 10 states, respectively. Conversely, when employing the single linkage method, one cluster emerges significantly larger, encompassing 47 states, while the remaining three clusters contain only one state each. Moreover, the cluster dendrograms exhibit disparity, illustrating that while the complete linkage method yields clusters of comparable size, the single linkage method tends to aggregate most states into a single cluster, as we can see in lower branches. The single linkage method produces a sizable cluster characterized by moderate values across all four features, namely murder rate, assault rate, rape rate, and urban population size. However, compared to the complete linkage method, the single linkage method does not effectively delineate distinct clusters when aiming to create 4 clusters. After adjusting the number of clusters to 9, which is deemed optimal according to the single linkage method, we observed considerable disparities in the sizes of the resulting clusters (7, 1, 4, 14, 10, 1, 10, 2, 1). Furthermore, the cluster plot revealed overlaps in the areas associated with each cluster. Consequently, our analysis suggests that, overall, the complete linkage method may perform better than the single linkage method in this scenario.