MLHW4\_JF

February 11, 2022

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ ggplot2 3.3.5 ✓ purrr 0.3.4  
## ✓ tibble 3.1.5 ✓ dplyr 1.0.7  
## ✓ tidyr 1.1.3 ✓ stringr 1.4.0  
## ✓ readr 2.0.1 ✓ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

## Loading required package: Rcpp

## ##   
## ## Amelia II: Multiple Imputation  
## ## (Version 1.8.0, built: 2021-05-26)  
## ## Copyright (C) 2005-2022 James Honaker, Gary King and Matthew Blackwell  
## ## Refer to http://gking.harvard.edu/amelia/ for more information  
## ##

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

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

Q’s for sneha: Qoes gpaq8totmin and gpaq11days have to be a factor? based on the question asked, I believe this is a continuous variable. For q4, if I do it one way the optimal number of clusters as 3. If i do it the other way I get the optimal number of clusters as 1. What should I do? also how do I use cbind in the first way ?

# Part I Setup: Data Cleaning, Iniital Linear Regression, and Data Partitioning,

# First Task: Perform basic data cleaning. Note which features are continuous, which are categorical and ensure they are being stored that way in your R dataset (That is, if categorical variables have been read-in as continuous variables, convert them to factors)  
  
class4 = read\_csv("class4\_p1.csv") %>%  
 janitor::clean\_names() %>%  
 rename(observation = x1) %>% #all the variables are cont so we have to convert them all to factors/provide levels except for healthy days (outcome) because we are doing a linear reg   
 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)  
 ) %>%   
 drop\_na() #dropping all the missing data. The observations don't start at 1 anymore bc it dropped some people

## New names:  
## \* `` -> ...1

## 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.

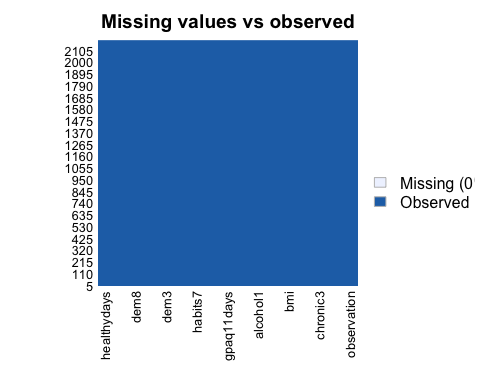
summary(class4)

## observation chronic1 chronic3 chronic4 bmi tobacco1 alcohol1  
## Min. :1324 1: 655 1: 250 1: 140 Min. : 9.09 1: 237 1: 67   
## 1st Qu.:1926 2:1540 2:1945 2:2055 1st Qu.:23.23 2: 118 2: 661   
## Median :2536 Median :26.57 3:1840 3:1467   
## Mean :2553 Mean :27.42   
## 3rd Qu.:3186 3rd Qu.:30.19   
## Max. :3809 Max. :99.83   
## gpaq8totmin gpaq11days habits5 habits7 agegroup dem3 dem4   
## Min. : 0.00 Min. :0.000 1: 661 1:283 1:167 1: 904 1: 476   
## 1st Qu.: 0.00 1st Qu.:2.000 2:1049 2:613 2:665 2:1291 2:1719   
## Median : 0.00 Median :5.000 3: 380 3:793 3:851   
## Mean : 44.91 Mean :4.241 4: 105 4:403 4:512   
## 3rd Qu.: 60.00 3rd Qu.:7.000 5:103   
## Max. :900.00 Max. :7.000   
## dem8 povertygroup healthydays   
## 1:1399 1:413 Min. : 0.00   
## 2: 796 2:365 1st Qu.:27.00   
## 3:371 Median :30.00   
## 4:357 Mean :26.42   
## 5:506 3rd Qu.:30.00   
## 6:183 Max. :30.00

missmap(class4, main = "Missing values vs observed")

## Warning: Unknown or uninitialised column: `arguments`.  
  
## Warning: Unknown or uninitialised column: `arguments`.

## Warning: Unknown or uninitialised column: `imputations`.



# First Task part2: Constructing my linear regression   
model1 = lm(healthydays ~ gpaq8totmin + gpaq11days + bmi, data = class4)  
summary(model1)

##   
## Call:  
## lm(formula = healthydays ~ gpaq8totmin + gpaq11days + bmi, data = class4)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -28.068 0.582 2.720 3.734 9.058   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.771e+01 7.967e-01 34.777 < 2e-16 \*\*\*  
## gpaq8totmin -6.288e-05 1.787e-03 -0.035 0.971938   
## gpaq11days 2.995e-01 6.206e-02 4.826 1.49e-06 \*\*\*  
## bmi -9.311e-02 2.550e-02 -3.652 0.000267 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 7.652 on 2191 degrees of freedom  
## Multiple R-squared: 0.01852, Adjusted R-squared: 0.01718   
## F-statistic: 13.78 on 3 and 2191 DF, p-value: 6.632e-09

confint(model1)

## 2.5 % 97.5 %  
## (Intercept) 26.144631871 29.269424165  
## gpaq8totmin -0.003567939 0.003442177  
## gpaq11days 0.177825342 0.421248804  
## bmi -0.143107441 -0.043104845

model2 = lm(healthydays ~ gpaq8totmin + gpaq11days + bmi + povertygroup + agegroup, data = class4)  
summary(model2)

##   
## Call:  
## lm(formula = healthydays ~ gpaq8totmin + gpaq11days + bmi + povertygroup +   
## agegroup, data = class4)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -30.3000 -0.1636 2.1742 3.8830 10.1018   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 27.720235 1.006081 27.553 < 2e-16 \*\*\*  
## gpaq8totmin -0.001685 0.001753 -0.961 0.33655   
## gpaq11days 0.262539 0.060732 4.323 1.61e-05 \*\*\*  
## bmi -0.065288 0.025113 -2.600 0.00939 \*\*   
## povertygroup2 1.047032 0.534871 1.958 0.05041 .   
## povertygroup3 2.983001 0.534107 5.585 2.63e-08 \*\*\*  
## povertygroup4 3.172107 0.539341 5.881 4.69e-09 \*\*\*  
## povertygroup5 3.569124 0.499280 7.149 1.19e-12 \*\*\*  
## povertygroup6 1.038708 0.674786 1.539 0.12387   
## agegroup2 -1.331862 0.665894 -2.000 0.04561 \*   
## agegroup3 -3.217298 0.657133 -4.896 1.05e-06 \*\*\*  
## agegroup4 -4.283027 0.679939 -6.299 3.61e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 7.441 on 2183 degrees of freedom  
## Multiple R-squared: 0.07535, Adjusted R-squared: 0.07069   
## F-statistic: 16.17 on 11 and 2183 DF, p-value: < 2.2e-16

confint(model2)

## 2.5 % 97.5 %  
## (Intercept) 25.747259020 29.693211133  
## gpaq8totmin -0.005122333 0.001752561  
## gpaq11days 0.143440221 0.381637925  
## bmi -0.114535037 -0.016040003  
## povertygroup2 -0.001878610 2.095941668  
## povertygroup3 1.935590802 4.030411508  
## povertygroup4 2.114431757 4.229782791  
## povertygroup5 2.590009596 4.548238311  
## povertygroup6 -0.284582127 2.361997783  
## agegroup2 -2.637713264 -0.026010551  
## agegroup3 -4.505970029 -1.928625481  
## agegroup4 -5.616422027 -2.949632192

# Second Task: Partition data into training and testing (use a 70/30 split)   
  
#Note to self: First I need to find the correlated predictors and remove them using the cor function and then the find correlation function. Set my cutoff to 0.9.  
class4\_num = class4 %>%   
 select(where(is.numeric)) #only selected observations,bmi, gpaq8,gpaq11,and healthydays  
  
correlations\_class4 <- cor(class4\_num, use = "complete.obs")  
highcorrelations <- findCorrelation(correlations\_class4, cutoff = 0.9) #high.correlations was empty so the data isn't correlated so the following dataset shouldn't have any variables.  
class4\_lowcorr = class4\_num[,-highcorrelations]  
  
#NTS: Now I need to center and scale my data using preprocess and then predict. I'll then set my seed to prevent constant randomization and drop the observation variable.  
  
preprocess = preProcess(class4\_num, method = c("center", "scale"))  
#Output pre-processed values  
transformed\_vals <- predict(preprocess, class4\_num)  
  
set.seed(1000)  
class4$observation <- NULL #copied from JS code  
  
train.index <- createDataPartition(class4$healthydays, p = 0.7, list = FALSE)  
  
class4\_train <- class4[train.index,]  
class4\_test <- class4[-train.index,]

# PART I: IMPLEMENTING A SIMPLE PREDICTION PIPELINE

## Question 1 and 2

#Q1 Task: Fit two prediction models using different subsets of the features in the training data. Features can overlap in the two models, but the feature sets should not be exactly the same across models. Clearly state which features were used in the two models.  
  
#Q2 Task: Apply both models within the test data and determine which model is the preferred prediction model using the appropriate evaluation metric(s).  
  
#NTS: If this was a logistic regression I would use a confusion matrix but since this is a linear regression I am using postResample.  
  
#NTS: First I'm tuning my hyperparameter using cross validation (through the train control function.) and doing a 3-fold cross-validation.  
control.settings <- trainControl(method = "cv", number = 3)  
  
#Now I'm running my train function  
set.seed(1000)  
model1\_again <- train(healthydays ~ gpaq8totmin + gpaq11days + bmi, data = class4, method = "glm", family = "gaussian", trControl = control.settings)  
  
model1\_pred <- predict(model1\_again, class4\_test)  
  
model1\_again

## Generalized Linear Model   
##   
## 2195 samples  
## 3 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (3 fold)   
## Summary of sample sizes: 1463, 1463, 1464   
## Resampling results:  
##   
## RMSE Rsquared MAE   
## 7.654338 0.01759874 4.972078

postResample(pred = model1\_pred, obs = class4\_test$healthydays)

## RMSE Rsquared MAE   
## 7.80046417 0.02783772 4.96932359

model2\_again <- train(healthydays ~ gpaq8totmin + gpaq11days + bmi + povertygroup + agegroup, data = class4, method = "glm", family = "gaussian", trControl = control.settings)  
  
model2\_pred <- predict(model2\_again, class4\_test)  
  
model2\_again

## Generalized Linear Model   
##   
## 2195 samples  
## 5 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (3 fold)   
## Summary of sample sizes: 1464, 1463, 1463   
## Resampling results:  
##   
## RMSE Rsquared MAE   
## 7.443614 0.07053439 4.77111

postResample(pred = model2\_pred, obs = class4\_test$healthydays)

## RMSE Rsquared MAE   
## 7.5042384 0.1019007 4.6532434

My first model includes 3 variables: BMI, gpaq8totmin(minutes of total physical activity on home chores on an average day) and gpaq11days (during the last 7 days, on how many days did you walk to get to and from places?). The root mean square error was 7.654. After running the postResample, my RSME became 7.800.

My second model includes these three variables plus poverty group (is your household’s annual income from all source: 1 = <100%, 2 = 100-199%, 3 = 200-399%, 4 = 400-599%, 5 = 600%, 6 = Don’t know) and agegroup (group 1 = 18-24, group 2 = 25-44, group 3 = 45-64, and group 4 = 65+). The root mean square error was 7.444. After running the postResample, my RSME became 7.502.

Since the Mean Square Error (RMSE) is a standard way to measure the error in our model, the larger the number the larger the error. I would recommend the second model which included 5 variables as the preferred final prediction model because it had a slightly lower RMSE.

## Question 3

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

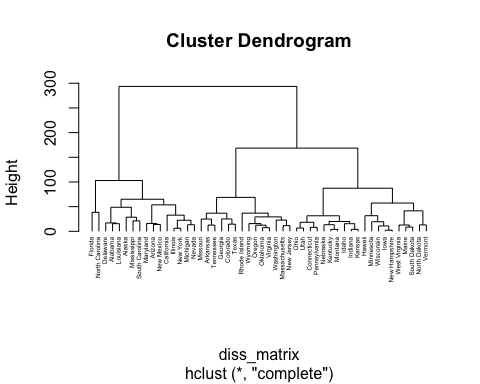
Q3 Answer: My final model may be useful in a low-income community where there may be high prevalence of chronic diseases and researchers would like to understand how predictors like poverty or physical activity levels impact an individuals typical amount of healthy days.

# PART II: CONDUCTING AN UNSUPERVISED ANALYSIS

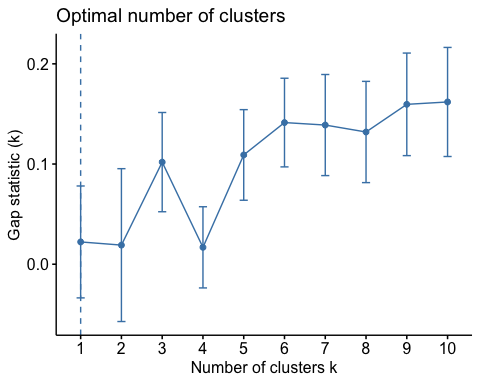
Using the dataset from the Group assignment Part 3 (USArrests), identify clusters using hierarchical analysis. Use an agglomerative algorithm for hierarchical clustering. Use a Euclidian distance measure to construct your dissimilarity matrix.

## Question 4

# FIRST WAY I DID IT BEFORE THE ALTERNATIVE WAY  
#Q4 Task: Conduct a hierarchical clustering analysis. Be sure to specify the linkage method used. Within your analysis, make sure you do both of the following:  
# 1. Determine the optimal number of clusters using a clear, data-driven strategy.  
# 2. Describe the composition of each cluster in terms of the original input features  
  
data(USArrests)  
  
# NTS: First I need to create Dissimilarity matrix  
diss\_matrix <- dist(USArrests, method = "euclidean")  
  
#NTS: Then hierarchical clustering using Complete Linkage  
clusters\_h <- hclust(diss\_matrix, method = "complete" )  
  
# Plot the obtained dendrogram  
plot(clusters\_h, cex = 0.4, hang = -1)



#NTS: Now I need to choose where to cut across my dendrogram to choose my number of clusters. In order to do that I have to creating a function to use within clusGap. I am now using an average linkage. After running the function, I will plot the gapstat to visualize it  
  
hclusCut <- function(x, k) list(cluster = cutree(hclust(dist(x, method = "euclidian"), method = "complete"), k = k))  
  
gap\_stat <- clusGap(USArrests, FUN = hclusCut, K.max = 10, B = 50)  
fviz\_gap\_stat(gap\_stat)



#NTS: Now, use the number of clusters from gap statistic to obtain cluster assignment for each observation  
clusters\_h3 = cutree(clusters\_h, k = 3)  
table(clusters\_h3)

## clusters\_h3  
## 1 2 3   
## 16 14 20

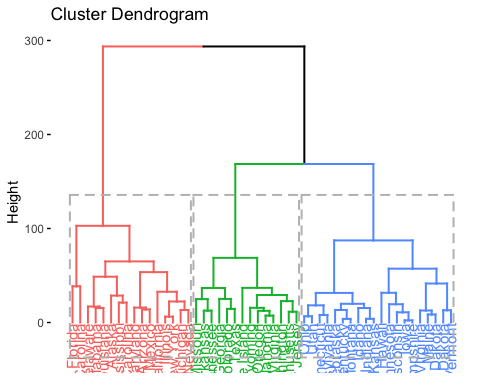
#Now I want to take those values and put it back onto my original dataset  
#input.feature.vals<-cbind(USArrests,cluster=clusters\_h3)  
  
#input.feature.vals %>%  
# group\_by(clusters\_h3) %>%  
# summarise\_all(mean)

#Question IV Task: Conduct a hierarchical clustering analysis. Be sure to specify the linkage method used. Within your analysis, make sure you do both of the following:  
# 1 Determine the optimal number of clusters using a clear, data-driven strategy.  
# 2 Describe the composition of each cluster in terms of the original input features  
  
  
#Alternatives for hierarchical clustering  
data(USArrests)  
  
clusters\_hcut <- hcut(USArrests, k = 3, hc\_func = "hclust", hc\_method = "complete", hc\_metric = "euclidian")  
  
clusters\_hcut$size

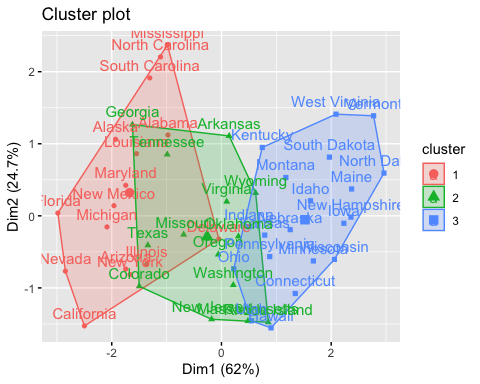
## [1] 16 14 20

fviz\_dend(clusters\_hcut, rect = TRUE)

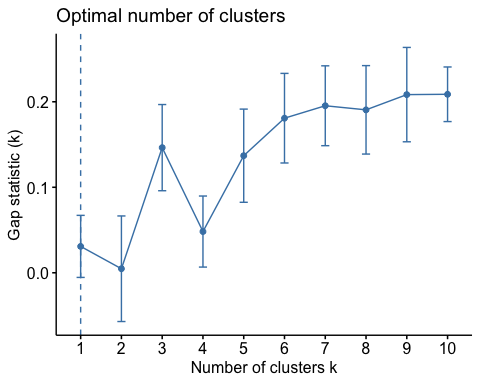
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =  
## "none")` instead.



fviz\_cluster(clusters\_hcut)



gap\_stat <- clusGap(USArrests, FUN = hcut, hc\_method = "complete", K.max = 10, B = 5)  
fviz\_gap\_stat(gap\_stat)



input.feature.vals <- cbind(USArrests,cluster = clusters\_hcut$cluster)  
  
input.feature.vals %>%  
 group\_by(cluster) %>%  
 summarise\_all(mean)

## # A tibble: 3 × 5  
## cluster Murder Assault UrbanPop Rape  
## <int> <dbl> <dbl> <dbl> <dbl>  
## 1 1 11.8 273. 68.3 28.4  
## 2 2 8.21 173. 70.6 22.8  
## 3 3 4.27 87.6 59.8 14.4

Q4 Answers: I used Euclidean as my distance metric and complete linkage so that I can use the distance between all of the different data points. After visualizing the gapstat, the optimal number of clusters was 3. In the first cluster there were 16 states, 14 states in the second cluster, and 20 states in the third cluster. In cluster one, the average murder (murder arrests per 100,000) was 11.8, the average assault (assault arrests per 100,000) was 273, the percent urban population was 68.3, and the average rape arrests (rape arrests per 100,000) was 28.4.

In cluster two, the average murder (murder arrests per 100,000) was 8.21, the average assault (assault arrests per 100,000) was 173, the percent urban population was 70.6, and the average rape arrests (rape arrests per 100,000) was 22.8.

In cluster three, the average murder (murder arrests per 100,000) was 4.27, the average assault (assault arrests per 100,000) was 87.6, the percent urban population was 59.8, and the average rape arrests (rape arrests per 100,000) was 14.4.

## Question 5

#Question 5 Task: 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.

Q5 Answer: Is there a relationship between number of COVID-19 deaths (outcome) and the percent of urban population within the clusters? One onsideration is that within the same cluster, some states may have a higher urban population percentage compared to others, which can bias the results of the analysis. For instance, New York and Alabama are in the same cluster but the two states have a big difference in their urban populations. Also the researchers may want to consider the differences in the access to care/affordability of care in the clusters, as this would be related to whether or not an individual may have received appropriate COVID treatment prior to death.