Final Project Appendix

2022-12-05

```r
###########################################
# This file is for data imputation and cleaning.
# MICE: Multiple Imputation with Chained Equations package
#install.packages('mice')
#install.packages('VIM')
library(mice)
library(VIM)

load("C:/Users/mbila/Downloads/exposome_NA.RData")

data <- read.csv('C:/Users/mbila/Documents/STAT 331 Final Project/Data/full_data_v2.csv')




## --------------- DEFINING CLEANING PROCESS -------------- ##


### Methods of cleaning
# https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4
# https://www.youtube.com/watch?v=MpnxwNXGV-E
# https://www.theanalysisfactor.com/multiple-imputation-in-a-nutshell/


# What are we counting as dirty data? N/A's only.
# Which covariates have NA's?
na <- which(lapply((lapply(data, is.na)), function(x) {length(x[x==TRUE])}) > 0)
dataRmv <- na.omit(data) # all obvs with no NA Values
md.pattern(data[, names(na)])
nrow(data) - nrow(dataRmv)

# 230 rows with some NA's
# 104 total values missing out of 1301*238 values
# => 17.7% of full data missing
# => 11.8% of missing rows contain NA for hs_wgtgain_None (154 rows)
wgt <- data[, 'hs_wgtgain_None']
length(wgt[!is.na(wgt)])

# Visualizing NA columns (EDA):
aggr_plot <- aggr(data, col=c('navyblue', 'red'),
                  numbers=TRUE,
                  sortVars=TRUE,
                  labels=names(data),
                  cex.axis=.7,
                 gap=3,
                 ylab=c("HISTOGRAM OF MISSING DATA", "PATTERN"))


## ------------ REMOVING WGT COVARIATE ------------ ##
## HOW to justify removing covariate?

# 1) Is distribution of NA's random or dependent on other covariates?
# 2) wgt is not related to raven score

# Plotting relationship between hs_correct_raven and wgt's complete values
toDel <- c()
for (i in 1:length(wgt)) {
  if (is.na(wgt[i])) {
    toDel <- c(toDel, i)
  }
}
dataShort <- data[-toDel, ]
wgtShort <- wgt[-toDel]

plot(wgtShort, dataShort[, 'hs_correct_raven']
     , ylab='Raven Score', xlab=codebook['hs_wgtgain_None', 'description'])

## No visible relationship in between Raven Score and Maternal Weight Gain
##  during pregnancy. No slope, no clustering of datapoints depending on
##  weightgain value.

## IS THIS ENOUGH TO KICK IT OUT?


## --------------- CLEANING OPTIONS -------------- ##

# List-wise Deletion: Removing rows with NA's, BIASED, WORST

# Mean/Mode Imputation: Replacing NAs with column mean/mode BIASED

# Multiple Imputation: Regressing NA covariate on rest, lowers Biasness

#   We need to prove MAR!!!!


## BEFORE ANYTHING, must check for wierd catagorical covariates which
# are stored as numerical


## --------------- CONVERTING CATEGORICAL COVARIATES -------------- ##


## Categorical Covariates labelled as numeric, Removing ID/X columns
if (length(which(colnames(data) == c('X', 'ID'))) != 0) {
  data <- data[, -which(colnames(data) == c('X', 'ID'))]
}
```

```r
# HELPER - Checking if function is true for all elements in a vector
floorCheck <- function(x) {
  for (i in 1:length(x)) {
    if (floor(x[i]) != x[i]) {
      return (FALSE)
    }
  }
  return (TRUE)
}

whichNumeric <- c()
for (i in 1:ncol(data)) {
  vec <- data[, i]
  if (is.numeric(vec)) {
    whichNumeric <- c(whichNumeric, i)
  }
}
dataNumeric <- data[, whichNumeric]
whichSus <- c()
for (i in 1:ncol(dataNumeric)) {
  vec <- dataNumeric[, i]
  vecNoNA <- na.omit(vec)
  if (floorCheck(vecNoNA)) {
    whichSus <- c(whichSus, i)
  }
}
dataSus <- na.omit(dataNumeric)[, whichSus]

# Ranges for each suspicious column
lapply(dataSus, range)

# Columns in dataSus are labelled as numeric, but are actually categorical
# SOLUTION: Replace numeric values with factored versions to
#   convert vector into categorical

# After modification of data, columns will be added back to original data
dataFinal <- data
whichFix <- which(colnames(data) %in% colnames(dataSus))
for (i in 1:ncol(data)) {
  if (i %in% whichFix) {
    dataFinal[, i] <- as.factor(dataFinal[, i])
  }
}



## --------------- CLEANING -------------- ##


# --/ List-wise Deletion
toDel <- c()
for (i in 1:nrow(dataFinal)) {
  if (length(which(is.na(dataFinal[i, ]))) > 0) {
    toDel <- c(toDel, i)
  }
}
dataFinalDel <- dataFinal[-toDel, ]
write.csv(dataFinalDel, file = 'C:/Users/mbila/Documents/STAT 331 Final Project/Data/full_clean_data_v0
          row.names = FALSE)

# --/ Mean/Mode Imputation
#### STEP 1: Acquiring column means/modes
modes <- rep(0, ncol(dataFinal))

getmode <- function(v) {
  unique(v)[which.max(tabulate(match(v, unique(v))))]
}

for (col in 1:ncol(dataFinal)) {
  curmode <- NULL

  ## Now, vec has no null values. Calculate and assign means.
  vec <- dataFinal[, col]

  if (is.numeric(vec)) {
    curmode <- mean(vec, na.rm = TRUE)
  } else {
    curmode <- getmode(vec)
  }
  modes[col] <- curmode
}

## Now, modes is complete

#### STEP 2: Assigning all NA's to same node value
cleandataFinal <- dataFinal

for (colnum in 1:ncol(cleandataFinal)) {
  col <- cleandataFinal[, colnum]
  for (rownum in 1:length(col)) {
    if (is.na(col[rownum])) {
      cleandataFinal[rownum, colnum] <- modes[colnum]
    }
  }
}

## Now, all NA's within one column hold same mode value
## Sanity check:
which(is.na(dataFinal)) # Should be non-zero vector
which(is.na(cleandataFinal)) # Should be 0

write.csv(cleandataFinal, file = 'C:/Users/mbila/Documents/STAT 331 Final Project/Data/full_clean_data_
          row.names = FALSE)

## Double check if u have all covariates, should be 241 not 237
## NOTE: # graph with MSPE, phi, lambda
```

```r
# --/ Multiple Imputation
# https://www.section.io/engineering-education/predictive-mean-matching/#solving-for-missing-values-usi

dataImp <- mice(dataFinal, m=5, method='pmm')
summary(dataImp)

# Checking Imputations for each column
dataImp$imp$hs_correct_raven

## PICKING BEST COLUMN
# For each imputed column, we replace column with
# one of the 5, one with best imputation

# JUSTIFY CHOOSING 5
dataF <- complete(dataImp, 5)

write.csv(dataF, file = 'C:/Users/mbila/Documents/STAT 331 Final Project/Data/full_clean_data_v2.csv',
          row.names = FALSE)
```

```r
load("/Users/srijanchaudhuri/Downloads/exposome_NA.RData")

df <- read.csv("/Users/srijanchaudhuri/Downloads/full_clean_data_v2.csv")

# Dropping useless covariates
drop <- c("ID","X")
all_data = df[,!(names(df) %in% drop)]

# Splits data
train_test <- function(data, p) {
  library(caret)
  set.seed(20893423)
  random_sample <- createDataPartition(data$hs_correct_raven, p = p,
                                       list = FALSE)# 80-20 split on all data
  training_data  <- data[random_sample, ]
  testing_data <- data[-random_sample, ]
  list(train = training_data, test = testing_data) # returning train and test
}

# Runs lasso on training data
selection_lasso <- function(train, covariates) {
  set.seed(331)
  library("dplyr")
  y <- train$hs_correct_raven
  x <- data.matrix(train[, covariates])
  library(glmnet)
  cv_model <- cv.glmnet(x, y, alpha = 1) # fitting a glm with cross-validation
  best_lambda <- cv_model$lambda.min # finding smallest lambda
  best_model <- glmnet(x, y, alpha = 1, lambda = best_lambda)
  v <- which(as.vector(coef(best_model))!=0)
  list(cov = covariates[v[2:length(v)] - 1], cvmod = cv_model)
}

# Fits a linear model for the given covariates and data
```

```r
make_linear <- function(train, covariates){
  frmla <- as.formula(paste("hs_correct_raven ~ ", # creating a formula
                            paste(covariates, collapse = "+"),
                            sep=""))
  model <- lm(frmla, data=train) # fitting linear model
  model
}

# Calculating root mean squared error
testing_rmse <- function(test, model, covariates) {
  library(Metrics)
  preds <- as.vector(predict(model,test[, covariates]))
  rmse(test$hs_correct_raven, preds) # using rmse metric from Metrics package
}

# Generates plots to test normality
error_normality_diagnosis <- function(model, main) {
  res <- resid(model) # extracting residuals
  stud <- res/(sigma(model)*sqrt(1-hatvalues(model))) # studentizing
  par(mfrow=c(2,1))
  hist(stud,breaks=12,
       probability=TRUE,xlim=c(-4,4),
       xlab="Studentized Residuals",
       main=paste("Distribution of Residuals ", main, sep="-"))
  grid <- seq(-3.5,3.5,by=0.05)
  lines(x=grid,y=dnorm(grid),col="blue") # generating N(0,1) plot
  qqnorm(stud)
  abline(0,1, col="red")
}

# Generates plots to test equal variance
error_variability_diagnosis <- function(model, main) {
  res <- resid(model) # extracting residuals
  fit <- fitted(model) # extracting fitted values
  plot(res~fit, xlab="Fitted Vals", ylab="Studentized Residuals",
       main=paste("Residuals vs Fitted", main, sep=" -"))
}

# Generates plots to test linearity
error_linearity_diagnosis <- function(model, train) {
  fit <- fitted(model) # extracting fitted values
  plot(train$hs_correct_raven~fit, xlab="Fitted Values", ylab="Actual Values",
       main="Actual vs Fitted")
}

# Generates plots to find out outliers
plot_outliers <- function(model, covariates) {
  X <- model.matrix(model) # extracting design matrix
  H <- X%*%solve(t(X)%*%X)%*%t(X)
  lev <- hatvalues(model) # calculating leverages
  hbar <- mean(lev)
  ids <- which(lev>2*hbar) # finding critical leverages
  res <- resid(model) # raw residuals
  stud <- res/(sigma(model)*sqrt(1-lev)) # studentizing
  p <- length(covariates)
  n <- nrow(X)
  jack <- stud*sqrt((n-p-2)/(n-p-1-stud^2)) # calculating jack-knife residuals
  plot(jack, ylab="Studentized Jackknife Residuals")
  points(jack[ids]~ids,col="red",pch=19)
  text(ids,jack[ids], labels=ids, col="yellow", cex= 0.6, pos=2)
}

# Splitting all the data into train and test
train_and_test <- train_test(all_data, p=0.8)
train <- train_and_test$train
test <- train_and_test$test

# Splitting train into train and select
select_and_train <- train_test(train, p=0.5)
train <- select_and_train$train
select <- train_and_test$test

# All covariates
covariates_full <- as.vector(
  codebook$variable_name[which(codebook$domain != "Phenotype") ])

# All covariates under Lifestyle
covariates_life <- as.vector(
  codebook$variable_name[which(codebook$family == "Lifestyle"
                               & codebook$domain != "Phenotype") ])

# All covariates under Pregnancy
covariates_preg <- as.vector(
  codebook$variable_name[which(codebook$period == "Pregnancy"
                               & codebook$family == "Lifestyle"
                               & codebook$domain != "Phenotype")])

# All covariates under Postnatal
covariates_post <- as.vector(
  codebook$variable_name[which(codebook$period == "Postnatal"
                               & codebook$family == "Lifestyle"
                               & codebook$domain != "Phenotype")])

# Performing variable selection to each subset
lasso_full <- selection_lasso(select, covariates_full)
lasso_life <- selection_lasso(select, covariates_life)
lasso_post <- selection_lasso(select, covariates_post)
lasso_preg <- selection_lasso(select, covariates_preg)
reduced_full <- lasso_full$cov
reduced_life <- lasso_life$cov
reduced_post <- lasso_post$cov
reduced_preg <- lasso_preg$cov
png("/Users/srijanchaudhuri/Desktop/Lasso_plots.png")
par(mar=c(6,6,6,6), mfrow=c(2,2))
plot(lasso_full$cvmod, main="Full Model")
plot(lasso_life$cvmod, main="Life Model")
plot(lasso_post$cvmod, main="Post-natal Model")
plot(lasso_preg$cvmod, main="Pregnancy Model")
dev.off()

# Fitting linear models to each subset
m_full <- make_linear(train, reduced_full)
m_life <- make_linear(train, reduced_life)
m_preg <- make_linear(train, reduced_preg)
m_post <- make_linear(train, reduced_post)

# Printing rmses
print(testing_rmse(test, m_full, reduced_full))
print(testing_rmse(test, m_life, reduced_life))
print(testing_rmse(test, m_preg, reduced_preg))
print(testing_rmse(test, m_post, reduced_post))

# Creating a results based table
results <- matrix(
  c(AIC(m_full), BIC(m_full),
    summary(m_full)$r.squared, summary(m_full)$adj.r.squared,
    testing_rmse(test, m_full, reduced_full)^2,
    testing_rmse(test, m_full, reduced_full),
    AIC(m_life), BIC(m_life),
    summary(m_life)$r.squared, summary(m_life)$adj.r.squared,
    testing_rmse(test, m_life, reduced_life)^2,
    testing_rmse(test, m_life, reduced_life),
    AIC(m_post), BIC(m_post),
    summary(m_post)$r.squared, summary(m_post)$adj.r.squared,
    testing_rmse(test, m_post, reduced_post)^2,
    testing_rmse(test, m_post, reduced_post),
    AIC(m_preg), BIC(m_preg),
    summary(m_preg)$r.squared, summary(m_preg)$adj.r.squared,
    testing_rmse(test, m_preg, reduced_preg)^2,
    testing_rmse(test, m_preg, reduced_preg)), nrow=4, ncol=6, byrow=TRUE)
colnames(results) <- c('AIC','BIC','Adjusted-R-Squared', 'R-Squared', 'MSE','RMSE')
rownames(results) <- c('Full','Life','Post-Natal', 'Pregnancy')
results <- as.table(results)
results

# Diagnosing linearity
png("/Users/srijanchaudhuri/Desktop/Linearity_plots.png")
par(mfrow=c(2,2))
error_linearity_diagnosis(m_full, train)
error_linearity_diagnosis(m_life, train)
error_linearity_diagnosis(m_post, train)
error_linearity_diagnosis(m_preg, train)
dev.off()

# Diagnosing normality
png("/Users/srijanchaudhuri/Desktop/Normality_full.png")
error_normality_diagnosis(m_full, main=" Full")
dev.off()
```

```r
png("/Users/srijanchaudhuri/Desktop/Normality_life.png")
error_normality_diagnosis(m_life, main=" Life")
dev.off()

png("/Users/srijanchaudhuri/Desktop/Normality_post.png")
error_normality_diagnosis(m_post, main=" Post")
dev.off()

png("/Users/srijanchaudhuri/Desktop/Normality_preg.png")
error_normality_diagnosis(m_preg, main=" Preg")
dev.off()

# diagnosing equal variance
png("/Users/srijanchaudhuri/Desktop/Residuals_full.png")
error_variability_diagnosis(m_full, " Full")
dev.off()

png("/Users/srijanchaudhuri/Desktop/Residuals_life.png")
error_variability_diagnosis(m_life, " Life")
dev.off()

png("/Users/srijanchaudhuri/Desktop/Residuals_post.png")
error_variability_diagnosis(m_post, " Post")
dev.off()

png("/Users/srijanchaudhuri/Desktop/Residuals_preg.png")
error_variability_diagnosis(m_preg, " Preg")
dev.off()

# Finding outliers
png("/Users/srijanchaudhuri/Desktop/Outliers.png")
plot_outliers(m_full, reduced_full)
dev.off()

# Plotting Fits
png("/Users/srijanchaudhuri/Desktop/Gr1.png")
plot(train$hs_correct_raven, main="Full")
lines(fitted(m_full),col="red",pch=19)
dev.off()

png("/Users/srijanchaudhuri/Desktop/Gr2.png")
plot(train$hs_correct_raven, main="Life")
lines(fitted(m_life),col="blue",pch=19)
dev.off()

png("/Users/srijanchaudhuri/Desktop/Gr3.png")
plot(train$hs_correct_raven, main="Post")
lines(fitted(m_post), col="green", pch=19)
dev.off()


png("/Users/srijanchaudhuri/Desktop/Gr4.png")
plot(train$hs_correct_raven, main="Preg")
```

```
lines(fitted(m_preg), col="yellow", pch=19)
dev.off()
```