```
# importing relevant packages
library(knitr)
library(ggplot2)
library(tidyverse)
library(dplyr)
library(glmnet)
library(vip)
library(broom)
library(grpreg)
library(car)
library(gam)
library(splines)
library(splines2)
library(mice)
library(miceadds)
library(ResourceSelection)
library(pROC)
library(VGAM)
library(doParallel)

# importing dataset with missing values
df <- read.csv("~/bst210-final-project/data/heart-2022-with-nans.csv")
dim(df)

# changing column capitalization
df <- df |> rename_with(tolower)

# selecting desired columns only
df <- df |> dplyr::select(-removedteeth, -hivtesting, -fluvaxlast12,
    -pneumovaxever,
                       -tetanuslast10tdap, -highrisklastyear, -covidpos,
                       -deaforhardofhearing, -blindorvisiondifficulty,
                       -difficultyconcentrating, -difficultywalking,
                       -difficultydressingbathing, -difficultyerrands,
                          -chestscan,
                       -lastcheckuptime)
dim(df)
colnames(df)

summarize_covariates <- function(data) {
  summary_table <- data.frame(
    Covariate = names(data),
    Type = sapply(data, function(x) {
      if (is.factor(x) || is.character(x)) "Categorical" else "Numeric"
    }),
    Levels = sapply(data, function(x) {if (is.factor(x) ||
        is.character(x)) {
      paste(unique(x), collapse = ", ")} else {NA}}),
    stringsAsFactors = FALSE)
  return(summary_table)
}

covariate_summary <- summarize_covariates(df)
covariate_summary

# replacing empty strings with NA
df[] <- lapply(df, function(x) replace(x, x == "", NA))

colnames(df)
lapply(df, unique)

# checking percent of missing values for each variable
percent_missing <- sapply(df, function(x) mean(is.na(x)) * 100)
kable(percent_missing[percent_missing > 0])
kable(percent_missing[percent_missing < 5 & percent_missing > 0])
kable(percent_missing[percent_missing > 5])

# checking for rows with height and weight but missing bmi
dim(df[!is.na(df$heightinmeters) & !is.na(df$weightinkilograms) &
    is.na(df$bmi),])

# imputing missing bmi with calculated bmi where valid
missing_bmi <- !is.na(df$heightinmeters) & !is.na(df$weightinkilograms) &
is.na(df$bmi)
sum(missing)
df$bmi[missing_bmi] <- df$weightinkilograms[missing_bmi]/(
df$heightinmeters[missing_bmi]^2)

# generating dataset for complete case analysis
# removing heightinmeters and weightinkilograms
# due to redundant information with bmi
df_no_hw <- df |> dplyr::select(-heightinmeters, -weightinkilograms)
df_complete <- df_no_hw[complete.cases(df_no_hw), ]
c(dim(df_no_hw), dim(df_complete))
write.csv(df_complete, file =
    "~/bst210-final-project/data/df_complete.csv",
row.names = FALSE)

# checking that missingness is not MCAR
# for those variables with missingness > 5%
# will check missingness and use imputation
percent_missing <- sapply(df_no_hw, function(x) mean(is.na(x)) * 100)
kable(percent_missing[percent_missing > 0])
kable(percent_missing[percent_missing > 5])

# variables to deal with are smokerstatus,
# ecigaretteusage, bmi, and alcoholdrinkers

# looking at missingness of the identified variables
par(mar = c(6, 6, 3, 2))
par(las = 2)
missing_pattern <- md.pattern(df[,c("smokerstatus", "ecigaretteusage",
    "bmi",
"alcoholdrinkers")])
missing_pattern_df <- as.data.frame(missing_pattern)
missing_pattern_sorted <-
    missing_pattern_df[order(as.numeric(rownames(missing_pattern_df)),
                                              decreasing = TRUE), ]
View(missing_pattern_sorted)

# getting percentages of the most observed missingness patterns
nrow(df[is.na(df$bmi) | is.na(df$smokerstatus) |
    is.na(df$ecigaretteusage) | is.na(df$alcoholdrinkers),])*100/nrow(df)
nrow(df[is.na(df$bmi) & !is.na(df$smokerstatus) &
    !is.na(df$ecigaretteusage) & !is.na(df$alcoholdrinkers),])*100/nrow(df)
nrow(df[is.na(df$bmi) & is.na(df$smokerstatus) &
    is.na(df$ecigaretteusage) & is.na(df$alcoholdrinkers),])*100/nrow(df)
nrow(df[!is.na(df$bmi) & is.na(df$smokerstatus) &
    is.na(df$ecigaretteusage) & is.na(df$alcoholdrinkers),])*100/nrow(df)
nrow(df[!is.na(df$bmi) & !is.na(df$smokerstatus) &
    !is.na(df$ecigaretteusage) & is.na(df$alcoholdrinkers),])*100/nrow(df)

# checking that missingness is not MCAR
# bmi
bmi_na <- df[is.na(df$bmi),]
df$bmi_indicator <- ifelse(is.na(df$bmi), 1, 0)
bmi_model <- glm(bmi_indicator ~ generalhealth + haddiabetes + hadstroke,
               family=binomial(), data = df)
summary(bmi_model)

# smokerstatus
smoking_na <- df[is.na(df$smokerstatus),]
df$smoker_indicator <- ifelse(is.na(df$smokerstatus), 1, 0)
smoker_model <- glm(smoker_indicator ~ generalhealth + mentalhealthdays +
    haddepressivedisorder,
                  family=binomial(), data = df)
summary(smoker_model)

# ecigaretteusage
ecig_na <- df[is.na(df$ecigaretteusage),]
df$ecig_indicator <- ifelse(is.na(df$ecigaretteusage), 1, 0)
ecig_model <- glm(ecig_indicator ~ generalhealth + mentalhealthdays +
    haddepressivedisorder,
                family=binomial(), data = df)
summary(ecig_model)

# alcoholdrinkers
alc_na <- df[is.na(df$alcoholdrinkers),]
df$alc_indicator <- ifelse(is.na(df$alcoholdrinkers), 1, 0)
alc_model <- glm(alc_indicator ~ generalhealth + haddiabetes +
    haddepressivedisorder, family=binomial(),
                data = df)
summary(alc_model)

# all variables of interest are identified as not being MCAR
# using EDA to investigate if missingness is MAR/MNAR

# bmi
# visualizing distribution of weight by bmi missingness
df_percentage <- df |>
  group_by(bmi_indicator, weightinkilograms) |>
  summarise(count = n(), .groups = 'drop') |>
  group_by(bmi_indicator) |>
  mutate(percentage = count / sum(count) * 100)

ggplot(df_percentage, aes(x = weightinkilograms, y = percentage)) +
  geom_bar(stat = "identity", color = "black") +
  facet_wrap(~bmi_indicator, scales = "free_y") +
  labs(title = "BMI missingness vs Weight", x = "Weight in kg", y =
      "Percentage (%)")

# visualizing distribution of height by bmi missingness
df_percentage <- df |>
  group_by(bmi_indicator, heightinmeters) |>
  summarise(count = n(), .groups = 'drop') |>
  group_by(bmi_indicator) |>
  mutate(percentage = count / sum(count) * 100)

ggplot(df_percentage, aes(x = heightinmeters, y = percentage)) +
  geom_bar(stat = "identity") +
  facet_wrap(~bmi_indicator, scales = "free_y") +
  labs(title = "BMI missingness vs Height", x = "Height in m", y =
      "Percentage (%)")

# percentage of individuals who did not report weight but reported height
# for those who did not report bmi but reported at least one of height or
    weight
nrow(df[df$bmi_indicator == 1 & is.na(df$weightinkilograms) &
    !is.na(df$heightinmeters),])*100/
  nrow(df[df$bmi_indicator == 1 &
      (!is.na(df$weightinkilograms)|!is.na(df$heightinmeters)),])

# percentage of individuals who did not report height but reported weight
# for those who did not report bmi but reported at least one of height or
    weight
nrow(df[df$bmi_indicator == 1 & !is.na(df$weightinkilograms) &
    is.na(df$heightinmeters),])*100/
  nrow(df[df$bmi_indicator == 1 &
      (!is.na(df$weightinkilograms)|!is.na(df$heightinmeters)),])

# smokerstatus
# visualizing distribution of bmi by smokerstatus missingness
```

```
df_percentage <- df |>
  group_by(smoker_indicator, bmi) |>
  summarise(count = n(), .groups = ’drop’) |>
  group_by(smoker_indicator) |>
  mutate(percentage = count / sum(count) * 100)

custom_labels <- c("0" = "Not Missing", "1" = "Missing")

ggplot(df_percentage, aes(x = bmi, y = percentage)) +
  geom_bar(stat = "identity", color = "black") +
  facet_wrap(˜smoker_indicator, scales = "free_y", labeller =
      labeller(smoker_indicator = custom_labels)) +
  labs(title = "Smokerstatus missingness vs BMI", x = "BMI", y =
      "Percentage (%)")

# visualizing distribution of e-cigarete use by smokerstatus missingness
df_percentage <- df |>
  group_by(smoker_indicator, ecigaretteusage) |>
  summarise(count = n(), .groups = ’drop’) |>
  group_by(smoker_indicator) |>
  mutate(percentage = count / sum(count) * 100)

ggplot(df_percentage, aes(x = ecigaretteusage, y = percentage)) +
  geom_bar(stat = "identity", color = "black") +
  facet_wrap(˜smoker_indicator, scales = "free_y", labeller =
      labeller(smoker_indicator = custom_labels)) +
  labs(title = "Smokerstatus missingness vs E-cigarette Use",
       x = "E-cigarette Use",
       y = "Percentage (%)") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),
        plot.title = element_text(hjust = 0.5))

# percentage of individuals who did not report e-cigarette use
# of those who did not report smokerstatus
nrow(df[df$smoker_indicator == 1 & is.na(df$ecigaretteusage),])*100/
  nrow(df[df$smoker_indicator == 1,])

# percentage of individuals who did not report e-cigarette use
# of those who did report smokerstatus
nrow(df[df$smoker_indicator == 0 & is.na(df$ecigaretteusage),])*100/
  nrow(df[df$smoker_indicator == 0,])

# visualizing distribution of alcohol use by smokerstatus missingness
df_percentage <- df |>
  group_by(smoker_indicator, alcoholdrinkers) |>
  summarise(count = n(), .groups = ’drop’) |>
  group_by(smoker_indicator) |>
  mutate(percentage = count / sum(count) * 100)

ggplot(df_percentage, aes(x = alcoholdrinkers, y = percentage)) +
  geom_bar(stat = "identity", color = "black") +
  facet_wrap(~smoker_indicator, scales = "free_y", labeller =
      labeller(smoker_indicator = custom_labels)) +
  labs(title = "Smokerstatus missingness vs Alcohol Use", x = "Alcohol
      Use", y = "Percentage (%)") +
  theme(plot.title = element_text(hjust = 0.5))

# percentage of individuals who did not report alcohol use
# of those who did not report smokerstatus
nrow(df[df$smoker_indicator == 1 &
    is.na(df$alcoholdrinkers),])*100/nrow(df[df$smoker_indicator == 1,])

# percentage of individuals who did not report alcohol use
# of those who did report smokerstatus
nrow(df[df$smoker_indicator == 0 &
    is.na(df$alcoholdrinkers),])*100/nrow(df[df$smoker_indicator == 0,])

# ecigaretteusage
# visualizing distribution of bmi by ecigaretteusage missingness
df_percentage <- df |>
  group_by(ecig_indicator, bmi) |>
  summarise(count = n(), .groups = ’drop’) |>
  group_by(ecig_indicator) |>
  mutate(percentage = count / sum(count) * 100)

ggplot(df_percentage, aes(x = bmi, y = percentage)) +
  geom_bar(stat = "identity", color = "black") +
  facet_wrap(~ecig_indicator, scales = "free_y") +
  labs(title = "Ecigaretteusage missingness vs BMI", x = "BMI", y =
      "Percentage (%)")

# visualizing distribution of alcohol use by ecigaretteusage missingness
df_percentage <- df |>
  group_by(ecig_indicator, alcoholdrinkers) |>
  summarise(count = n(), .groups = ’drop’) |>
  group_by(ecig_indicator) |>
  mutate(percentage = count / sum(count) * 100)

ggplot(df_percentage, aes(x = alcoholdrinkers, y = percentage)) +
  geom_bar(stat = "identity", color = "black") +
  facet_wrap(~ecig_indicator, scales = "free_y", labeller =
      labeller(ecig_indicator = custom_labels)) +
  labs(title = "Ecigaretteusage missingness vs Alcohol Use", x = "Alcohol
      Use", y = "Percentage (%)") +
  theme(plot.title = element_text(hjust = 0.5))

# percentage of individuals who did not report alcohol use
# of those who did not report ecigaretteusage
nrow(df[df$ecig_indicator == 1 &
    is.na(df$alcoholdrinkers),])*100/nrow(df[df$ecig_indicator == 1,])

# percentage of individuals who did not report alcohol use
# of those who did report ecigaretteusage
nrow(df[df$ecig_indicator == 0 &
    is.na(df$alcoholdrinkers),])*100/nrow(df[df$ecig_indicator == 0,])

# visualizing distribution of smokerstatus by ecigaretteusage missingness
df_percentage <- df |>
  group_by(ecig_indicator, smokerstatus) |>
  summarise(count = n(), .groups = 'drop') |>
  group_by(ecig_indicator) |>
  mutate(percentage = count / sum(count) * 100)

ggplot(df_percentage, aes(x = smokerstatus, y = percentage)) +
  geom_bar(stat = "identity", color = "black") +
  facet_wrap(~ecig_indicator, scales = "free_y", labeller =
      labeller(ecig_indicator = custom_labels)) +
  labs(title = "Ecigaretteusage missingness vs Smokerstatus", x =
      "Smokerstatus", y = "Percentage (%)") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),
        plot.title = element_text(hjust = 0.5))

# percentage of individuals who did not report smokerstatus
# of those who did not report ecigaretteusage
nrow(df[df$ecig_indicator == 1 &
    is.na(df$smokerstatus),])*100/nrow(df[df$ecig_indicator == 1,])

# percentage of individuals who did not report smokerstatus
# of those who did report ecigaretteusage
nrow(df[df$ecig_indicator == 0 &
    is.na(df$smokerstatus),])*100/nrow(df[df$ecig_indicator == 0,])

# alcoholdrinkers
# visualizing distribution of bmi by alcoholdrinkers missingness
df_percentage <- df |>
  group_by(alc_indicator, bmi) |>
  summarise(count = n(), .groups = 'drop') |>
  group_by(alc_indicator) |>
  mutate(percentage = count / sum(count) * 100)

ggplot(df_percentage, aes(x = bmi, y = percentage)) +
  geom_bar(stat = "identity", color = "black") +
  facet_wrap(~alc_indicator, scales = "free_y") +
  labs(title = "Alcoholdrinkers missingness vs BMI", x = "BMI", y =
      "Percentage (%)")

# visualizing distribution of ecigaretteusage by alcoholdrinkers
    missingness
df_percentage <- df |>
  group_by(alc_indicator, ecigaretteusage) |>
  summarise(count = n(), .groups = 'drop') |>
  group_by(alc_indicator) |>
  mutate(percentage = count / sum(count) * 100)
```

```
ggplot(df_percentage, aes(x = ecigaretteusage, y = percentage)) +
  geom_bar(stat = "identity", color = "black") +
  facet_wrap(~alc_indicator, scales = "free_y", labeller =
      labeller(alc_indicator = custom_labels)) +
  labs(title = "Alcoholdrinkers missingness vs Ecigaretteusage", x =
      "Ecigaretteusage", y = "Percentage (%)") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),
        plot.title = element_text(hjust = 0.5))

# percentage of individuals who did not report ecigaretteusage
# of those who did not report alcoholdrinkers
nrow(df[df$alc_indicator == 1 &
    is.na(df$ecigaretteusage),])*100/nrow(df[df$alc_indicator == 1,])

# percentage of individuals who did not report ecigaretteusage
# of those who did report alcoholdrinkers
nrow(df[df$alc_indicator == 0 &
    is.na(df$ecigaretteusage),])*100/nrow(df[df$alc_indicator == 0,])

# visualizing distribution of smokerstatus by alcoholdrinkers missingness
df_percentage <- df |>
  group_by(alc_indicator, smokerstatus) |>
  summarise(count = n(), .groups = ’drop’) |>
  group_by(alc_indicator) |>
  mutate(percentage = count / sum(count) * 100)

ggplot(df_percentage, aes(x = smokerstatus, y = percentage)) +
  geom_bar(stat = "identity", color = "black") +
  facet_wrap(~alc_indicator, scales = "free_y", labeller =
      labeller(alc_indicator = custom_labels)) +
  labs(title = "Alcoholdrinkers missingness vs Smokerstatus", x =
      "Smokerstatus", y = "Percentage (%)") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),
        plot.title = element_text(hjust = 0.5))

# percentage of individuals who did not report smokerstatus
# of those who did not report alcoholdrinkers
nrow(df[df$alc_indicator == 1 &
    is.na(df$smokerstatus),])*100/nrow(df[df$alc_indicator == 1,])

# percentage of individuals who did not report smokerstatus
# of those who did report alcoholdrinkers
nrow(df[df$alc_indicator == 0 &
    is.na(df$smokerstatus),])*100/nrow(df[df$alc_indicator == 0,])

# dropping indicator columns from dataset
# and dropping height and weight from dataset
# and cleaning dataset for imputation
df <- df_no_hw
sapply(df, unique)
```

```
df$state <- factor(df$state)
df$sex <- factor(df$sex, levels = c("Male", "Female"))
df$generalhealth <- factor(df$generalhealth, levels = c("Poor", "Fair",
    "Good",
                                                   "Very good",
                                                       "Excellent"))
df$physicalactivities <- ifelse(df$physicalactivities == "Yes", 1,
                             ifelse(df$physicalactivities == "No", 0, NA))
df$hadheartattack <- ifelse(df$hadheartattack == "Yes", 1,
                         ifelse(df$hadheartattack == "No", 0, NA))
df$hadangina <- ifelse(df$hadangina == "Yes", 1,
                     ifelse(df$hadangina == "No", 0, NA))
df$hadstroke <- ifelse(df$hadstroke == "Yes", 1,
                     ifelse(df$hadstroke == "No", 0, NA))
df$hadasthma <- ifelse(df$hadasthma == "Yes", 1,
                     ifelse(df$hadasthma == "No", 0, NA))
df$hadskincancer <- ifelse(df$hadskincancer == "Yes", 1,
                         ifelse(df$hadskincancer == "No", 0, NA))
df$hadcopd <- ifelse(df$hadcopd == "Yes", 1,
                   ifelse(df$hadcopd == "No", 0, NA))
df$haddepressivedisorder <- ifelse(df$haddepressivedisorder == "Yes", 1,
                                ifelse(df$haddepressivedisorder == "No",
                                    0, NA))
df$hadkidneydisease <- ifelse(df$hadkidneydisease == "Yes", 1,
                            ifelse(df$hadkidneydisease == "No", 0, NA))
df$hadarthritis <- ifelse(df$hadarthritis == "Yes", 1,
                        ifelse(df$hadarthritis == "No", 0, NA))
df$haddiabetes <- factor(df$haddiabetes, levels = c("No",
                                                "No, pre-diabetes or
                                                   borderline diabetes",
                                                "Yes, but only during
                                                   pregnancy (female)",
                                                "Yes"))
df$smokerstatus <- factor(df$smokerstatus, levels = c("Never smoked",
                                                  "Former smoker",
                                                  "Current smoker - now
                                                     smokes some days",
                                                  "Current smoker - now
                                                     smokes every day"))
df$ecigaretteusage <- factor(df$ecigaretteusage, levels = c("Never used
    e-cigarettes in my entire life",
                                                       "Not at all (right
                                                           now)",
                                                       "Use them some
                                                           days",
                                                       "Use them every
                                                           day"))
df$raceethnicitycategory <- factor(df$raceethnicitycategory)
df$agecategory <- factor(df$agecategory)
df$alcoholdrinkers <- ifelse(df$alcoholdrinkers == "Yes", 1,
                          ifelse(df$alcoholdrinkers == "No", 0, NA))
sapply(df, unique)
str(df)

# checking bmi - remove everything that is above 40 and below 18.5
par(mar = c(5, 6, 4, 2))
par(mgp = c(4, 1, 0))

dim(df)

# bmi - initial
bmi_clean <- df$bmi[!is.na(df$bmi)]
hist(bmi_clean, main = "Histogram of BMI", xlab = "BMI",
     col = "blue", border = "black", breaks = 100)
abline(v = 40, col = "red", lwd = 2, lty = 2)
abline(v = 18.5, col = "red", lwd = 2, lty = 2)
axis(side = 1, at = c(18.5, 40), labels = c("BMI = 18.5", "BMI = 40"),
   col.axis = "red")

100*sum(df$bmi < 18.5, na.rm = TRUE)/sum(!is.na(df$bmi))
100*sum(df$bmi >= 40, na.rm = TRUE)/sum(!is.na(df$bmi))
bmi_excluded <- sum(df$bmi < 18.5 | df$bmi >= 40, na.rm = TRUE)
bmi_excluded
bmi_more_than_50 <- sum(df$bmi > 50, na.rm = TRUE)
bmi_more_than_50
bmi_more_than_100 <- sum(df$bmi > 100, na.rm = TRUE)
bmi_more_than_100

# sleephours - initial
sleephours_clean <- df$sleephours[!is.na(df$sleephours)]
hist(sleephours_clean, main = "Histogram of Sleephours", xlab =
   "Sleephours",
     col = "blue", border = "black", breaks = 30)
abline(v = 12, col = "red", lwd = 2, lty = 2)
abline(v = 3, col = "red", lwd = 2, lty = 2)
axis(side = 1, at = c(3,12), labels = c("SH = 3", "SH = 12"), col.axis =
   "red")

100*sum(df$sleephours < 3, na.rm = TRUE)/sum(!is.na(df$sleephours))
100*sum(df$sleephours > 12, na.rm = TRUE)/sum(!is.na(df$sleephours))
sleephours_more_than_3 <- sum(df$sleephours < 3, na.rm = TRUE)
sleephours_more_than_3
sleephours_more_than_12 <- sum(df$sleephours > 12, na.rm = TRUE)
sleephours_more_than_12
sleephours_excluded <- sum(df$sleephours < 3 | df$sleephours > 12, na.rm
   = TRUE)
sleephours_excluded

# creating new df which includes only complete cases and excludes certain
   values of bmi and sleephours
df_upd <- df[complete.cases(df), ]
```

```
dim(df_upd)
df_upd <- df_upd |> filter((bmi >= 18.5 & bmi < 40)) |> filter(sleephours
   >= 3 & sleephours <= 12)
dim(df_upd)

# bmi - cleaned
hist(df_upd$bmi, main = "Histogram of cleaned BMI", xlab = "BMI",
     col = "blue", border = "black", breaks = 30, las = 1)

# sleephours - cleaned
hist(df_upd$sleephours, main = "Histogram of cleaned sleephours", xlab =
   "Sleephours",
     col = "blue", border = "black", breaks = 10, las = 1)

100*dim(df_upd)[1]/dim(df)[1]
dim(df_upd)
colnames(df_upd)
write.csv(df_upd, "~/bst210-final-project/data/df_complete.csv",
   row.names = FALSE)

# all four variables of interest are likely MNAR
# so we will proceed with imputation
methods <- c(state="", sex="", generalhealth="",
            physicalhealthdays="", mentalhealthdays="",
            physicalactivities="", sleephours="",
            hadheartattack="", hadangina="", hadstroke="",
            hadasthma="", hadskincancer="", hadcopd="",
            haddepressivedisorder="", hadkidneydisease="",
            hadarthritis="", haddiabetes="",
            smokerstatus="pmm", ecigaretteusage="pmm",
            raceethnicitycategory="", agecategory="",
            bmi="pmm", alcoholdrinkers="pmm")

nc <- detectCores() - 1
cl <- makeCluster(nc)
registerDoParallel(cl)

temp_data <- mice(df, m = 5, method = methods, seed = 210)
saveRDS(temp_data, file = "imputed_data.rds")
summary(temp_data)

stopCluster(cl)
stopImplicitCluster()

# verifying imputation
temp_check <- complete(temp_data, action=1)
sapply(temp_check, unique)
str(temp_check)
unique(df$alcoholdrinkers)
unique(df$smokerstatus)
unique(df$ecigaretteusage)
```

```
# all missingness is below 5%
percent_missing <- sapply(temp_check, function(x) mean(is.na(x)) * 100)
kable(percent_missing[percent_missing > 0])
kable(percent_missing[percent_missing < 5 & percent_missing > 0])
kable(percent_missing[percent_missing > 5])

dim(df[complete.cases(df), ])
dim(temp_check[complete.cases(temp_check), ])

# view the individual datasets using
# complete(temp_data, action=1), for 1-5
# to use the 5 imputed datasets use the following
# fitted_model = with(temp_data, model_formula)
# results = pool(fitted_model)
# summary(results)

imputed_data <- readRDS("~/bst210-final-project/imputed_data.rds")
data <- read.csv("~/bst210-final-project/data/df_complete.csv")
colnames(data)
dim(data)
colSums(is.na(data))

# setting factors with adequate reference levels
data <- data |>
  mutate(across(where(~ all(. %in% c("Yes", "No"))), ~ ifelse(. == "Yes",
      1, 0))) |>
  mutate(sex = ifelse(sex == "Female", 1, 0)) |>
  mutate(
    agecategory = as.factor(agecategory),
    ecigaretteusage = factor(ecigaretteusage, levels = c(
      "Never used e-cigarettes in my entire life",
      "Not at all (right now)",
      "Use them some days",
      "Use them every day")),
    smokerstatus = factor(smokerstatus, levels = c(
      "Never smoked",
      "Former smoker",
      "Current smoker - now smokes some days",
      "Current smoker - now smokes every day")),
    generalhealth = factor(generalhealth, levels = c(
      "Poor", "Fair", "Good", "Very good", "Excellent")),
    haddiabetes = factor(haddiabetes, levels = c(
      "No",
      "No, pre-diabetes or borderline diabetes",
      "Yes, but only during pregnancy (female)",
      "Yes")),
    raceethnicitycategory = factor(raceethnicitycategory, levels =
        unique(raceethnicitycategory))
  ) |>
  mutate(smokerstatus = relevel(smokerstatus, ref = "Never smoked"),
        raceethnicitycategory = relevel(raceethnicitycategory, ref =
            "White only, Non-Hispanic"))
str(data)

# substituting state information with region information
state_to_region <- list(
  Northeast = c("Connecticut", "Maine", "Massachusetts", "New Hampshire",
              "New Jersey", "New York", "Pennsylvania", "Rhode Island",
                  "Vermont"),
  Midwest = c("Illinois", "Indiana", "Iowa", "Kansas", "Michigan",
     "Minnesota",
            "Missouri", "Nebraska", "North Dakota", "Ohio", "South
                Dakota", "Wisconsin"),
  South = c("Alabama", "Arkansas", "Delaware", "District of Columbia",
     "Florida",
           "Georgia", "Kentucky", "Louisiana", "Maryland", "Mississippi",
           "North Carolina", "Oklahoma", "South Carolina", "Tennessee",
              "Texas",
           "Virginia", "West Virginia"),
  West = c("Alaska", "Arizona", "California", "Colorado", "Hawaii",
     "Idaho",
          "Montana", "Nevada", "New Mexico", "Oregon", "Utah",
             "Washington",
          "Wyoming"),
  Other = c("Guam", "Puerto Rico", "Virgin Islands"))

classify_region <- function(state) {
  region <- names(state_to_region)[sapply(state_to_region,
     function(region_states) state %in% region_states)]
  ifelse(length(region) > 0, region, NA)
}

# amending dataset for complete case analysis
data <- data |> mutate(region = as.factor(unname(sapply(state,
   classify_region))))
str(data)

data$bmi_category <- cut(data$bmi, breaks = c(18.5, 25.0, 30.0, 40.0),
                      labels = c("normalweight", "overweight", "obese"),
                      right = FALSE)

table(data$bmi_category)
lapply(data, unique)
str(data)

# amending imputed data for further analysis
add_region_column <- function(dataset) {
  dataset$region <- sapply(dataset$state, classify_region)
  return(dataset)
}
```

```
categorize_bmi <- function(dataset) {
  dataset$bmi_category <- cut(
    dataset$bmi,
    breaks = c(18.5, 25.0, 30.0, 40.0),
    labels = c("normalweight", "overweight", "obese"),
    right = FALSE
  )
  return(dataset)
}

dim(complete(imputed_data, action = 1))

imputed_data_long <- complete(imputed_data, "long", include = TRUE)
imputed_data_long <- imputed_data_long |>
  group_by(.imp) |>
  group_modify(˜ add_region_column(.x)) |>
  group_modify(˜ categorize_bmi(.x)) |>
  group_modify(˜ {
    .x$sleephours[.x$sleephours < 3 | .x$sleephours > 12] <- NA
    return(.x)
  })
imputed_data_final <- as.mids(imputed_data_long)

row_counts <- c()
complete_cases <- c()
for (i in 1:5) {
  imputed_dataset <- complete(imputed_data_final, action = i)
  row_counts[i] <- dim(imputed_dataset)[1]
  complete_cases[i] <-
      dim(imputed_dataset[complete.cases(imputed_dataset), ])[1]
}
row_counts
complete_cases
dim(data)

temp_check <- complete(imputed_data_final, action=1)
colnames(temp_check)
table(temp_check$sleephours)
table(temp_check$bmi_category)

# LOGISTIC
data_selection <- data |> dplyr::select(-c("state", "sex",
    "raceethnicitycategory",
                                          "agecategory", "bmi"))
dim(data_selection)
colnames(data_selection)
colSums(is.na(data_selection))

# forward, backward, step-wise selection
full_model_logistic <- glm(hadheartattack ˜ ., family = binomial(), data
    = data_selection)
```

```
summary(full_model_logistic)
length(all.vars(formula(full_model_logistic))[-1])
null_model_logistic <- glm(hadheartattack ˜ 1, family = binomial(), data
   = data_selection)

# model select via forward:
forward_model_logistic <- step(null_model_logistic, scope = list(lower =
   null_model_logistic, upper = full_model_logistic), direction =
   c("forward"), trace=0)
summary(forward_model_logistic)
all.vars(formula(forward_model_logistic))
length(all.vars(formula(forward_model_logistic))[-1])

# model select via backward:
backward_model_logistic <- step(full_model_logistic,
   direction=c("backward"), trace=0)
summary(backward_model_logistic)
all.vars(formula(backward_model_logistic))
length(all.vars(formula(backward_model_logistic))[-1])

# model select via step-wise:
step_model_logistic <- step(null_model_logistic, scope = list(lower =
   null_model_logistic, upper = full_model_logistic), direction=c("both"))
summary(step_model_logistic)
all.vars(formula(step_model_logistic))
length(all.vars(formula(step_model_logistic))[-1])

dif_logistic <- setdiff(all.vars(formula(full_model_logistic))[-1],
   all.vars(formula(step_model_logistic))[-1])
dif_logistic

# checking for multicollinearity
vif(step_model_logistic)

# LASSO

# prepare X matrix (minus death) for input to glmnet
x <- model.matrix(hadheartattack ˜ ., data = data_selection)[, -1]
y <- data_selection$hadheartattack
colnames(x)
ncol(x)

lambda_grid <- exp(seq(log(0.001), log(10), length.out = 100))
lasso.logistic = glmnet(x, y, alpha = 1, family = "binomial", lambda =
   lambda_grid)
vip(lasso.logistic, num_features = ncol(x), geom = "point",
   include_type=TRUE)
par(mfrow=c(1,2))
plot(lasso.logistic)
cv.lasso <- cv.glmnet(x, y, alpha = 1, family = "binomial")
lambda_min_lasso <- cv.lasso$lambda.min
```

```
lambda_1se_lasso <- cv.lasso$lambda.1se
plot(cv.lasso)
coef(cv.lasso,s=lambda_min_lasso)
coef(cv.lasso,s=lambda_1se_lasso)

dim(data)
colnames(data)
colSums(is.na(data))

# initial model
logistic_model_1 <- glm(hadheartattack ~ generalhealth +
   physicalactivities +
                       hadangina + hadstroke + hadasthma + hadskincancer +
                       hadcopd + haddepressivedisorder + hadkidneydisease
                           +
                       hadarthritis + haddiabetes + smokerstatus +
                           ecigaretteusage +
                       alcoholdrinkers + bmi_category, family =
                           binomial(),
                     data = data)

summary(logistic_model_1)
AIC(logistic_model_1)
vif(logistic_model_1)

# add agecategory to the basic model
logistic_model_2 <- glm(hadheartattack ~ generalhealth +
   physicalactivities +
                       hadangina + hadstroke + hadasthma + hadskincancer +
                       hadcopd + haddepressivedisorder + hadkidneydisease
                           +
                       hadarthritis + haddiabetes + smokerstatus +
                           ecigaretteusage +
                       alcoholdrinkers + bmi_category + agecategory,
                           family = binomial(),
                     data = data)

# add sex to the model with agecategory
logistic_model_3 <- glm(hadheartattack ~ generalhealth +
   physicalactivities +
                       hadangina + hadstroke + hadasthma + hadskincancer +
                       hadcopd + haddepressivedisorder + hadkidneydisease
                           +
                       hadarthritis + haddiabetes + smokerstatus +
                           ecigaretteusage +
                       alcoholdrinkers + bmi_category + agecategory +
                           sex, family = binomial(),
                     data = data)

# add race and ethnicity to the model with agecategory and sex
logistic_model_4 <- glm(hadheartattack ~ generalhealth +
  physicalactivities +
            hadangina + hadstroke + hadasthma + hadskincancer +
            hadcopd + haddepressivedisorder + hadkidneydisease
              +
            hadarthritis + haddiabetes + smokerstatus +
              ecigaretteusage +
            alcoholdrinkers + bmi_category + agecategory + sex
              + raceethnicitycategory,
           family = binomial(), data = data)
vif(logistic_model_4)

logOR.scale <- summary(logistic_model_4)$coefficients
OR_complete <- exp(logOR.scale[, "Estimate"])
lower_CI_complete <- exp(logOR.scale[, "Estimate"] - 1.96 * logOR.scale[,
  "Std. Error"])
upper_CI_complete <- exp(logOR.scale[, "Estimate"] + 1.96 * logOR.scale[,
  "Std. Error"])
OR_results_complete <- data.frame(
 Term = rownames(logOR.scale),
 OR_Complete = OR_complete,
 Lower_95_CI_Complete = lower_CI_complete,
 Upper_95_CI_Complete = upper_CI_complete
)
OR_results_complete

# display results
summary(logistic_model_2)
summary(logistic_model_3)
summary(logistic_model_4)

# perform likelihood ratio tests
lrt_result_2 <- anova(logistic_model_1, logistic_model_2, test = "LRT")
lrt_result_3 <- anova(logistic_model_2, logistic_model_3, test = "LRT")
lrt_result_4 <- anova(logistic_model_3, logistic_model_4, test = "LRT")
lrt_result_2
lrt_result_3
lrt_result_4

# check AIC
AIC(logistic_model_1)
AIC(logistic_model_2)
AIC(logistic_model_3)
AIC(logistic_model_4)

# change in AIC
100*(AIC(logistic_model_1) - AIC(logistic_model_4))/AIC(logistic_model_1)

# assessment of effect modifications
# interaction term: sex * raceethnicitycategory
logistic_model_5 <- glm(hadheartattack ˜ generalhealth +
  physicalactivities +
                        hadangina + hadstroke + hadasthma + hadskincancer +
                        hadcopd + haddepressivedisorder + hadkidneydisease
                            +
                        hadarthritis + haddiabetes + smokerstatus +
                            ecigaretteusage +
                        alcoholdrinkers + bmi_category + agecategory + sex
                            + raceethnicitycategory*sex,
                      family = binomial(), data = data)
summary(logistic_model_5)

# interaction term: sex * agecategory
logistic_model_6 <- glm(hadheartattack ~ generalhealth +
    physicalactivities +
                        hadangina + hadstroke + hadasthma + hadskincancer +
                        hadcopd + haddepressivedisorder + hadkidneydisease
                            +
                        hadarthritis + haddiabetes + smokerstatus +
                            ecigaretteusage +
                        alcoholdrinkers + bmi_category + agecategory*sex +
                            sex + raceethnicitycategory,
                      family = binomial(), data = data)
summary(logistic_model_6)

# interaction term: sex * bmi_category
logistic_model_7 <- glm(hadheartattack ~ generalhealth +
    physicalactivities +
                        hadangina + hadstroke + hadasthma + hadskincancer +
                        hadcopd + haddepressivedisorder + hadkidneydisease
                            +
                        hadarthritis + haddiabetes + smokerstatus +
                            ecigaretteusage +
                        alcoholdrinkers + bmi_category*sex + agecategory +
                            sex + raceethnicitycategory,
                      family = binomial(), data = data)
summary(logistic_model_7)

# interaction term: sex * smokerstatus
logistic_model_8 <- glm(hadheartattack ~ generalhealth +
    physicalactivities +
                        hadangina + hadstroke + hadasthma + hadskincancer +
                        hadcopd + haddepressivedisorder + hadkidneydisease
                            +
                        hadarthritis + haddiabetes + smokerstatus*sex +
                            ecigaretteusage +
                        alcoholdrinkers + bmi_category + agecategory + sex
                            + raceethnicitycategory,
                      family = binomial(), data = data)
summary(logistic_model_8)

# interaction term: sex * ecigaretteusage
logistic_model_9 <- glm(hadheartattack ~ generalhealth +
    physicalactivities +
                        hadangina + hadstroke + hadasthma + hadskincancer +
                        hadcopd + haddepressivedisorder + hadkidneydisease
                            +
                        hadarthritis + haddiabetes + smokerstatus +
                            ecigaretteusage*sex +
                        alcoholdrinkers + bmi_category + agecategory + sex
                            + raceethnicitycategory,
                      family = binomial(), data = data)
summary(logistic_model_9)

# interaction term: sex * haddiabetes
logistic_model_10 <- glm(hadheartattack ˜ generalhealth +
    physicalactivities +
                         hadangina + hadstroke + hadasthma + hadskincancer
                             +
                         hadcopd + haddepressivedisorder +
                             hadkidneydisease +
                         hadarthritis + haddiabetes*sex + smokerstatus +
                             ecigaretteusage +
                         alcoholdrinkers + bmi_category + agecategory +
                             sex + raceethnicitycategory,
                       family = binomial(), data = data)
summary(logistic_model_10)

# interaction term: sex * generalhealth
logistic_model_11 <- glm(hadheartattack ˜ generalhealth*sex +
    physicalactivities +
                         hadangina + hadstroke + hadasthma + hadskincancer
                             +
                         hadcopd + haddepressivedisorder +
                             hadkidneydisease +
                         hadarthritis + haddiabetes + smokerstatus +
                             ecigaretteusage +
                         alcoholdrinkers + bmi_category + agecategory +
                             sex + raceethnicitycategory,
                       family = binomial(), data = data)
summary(logistic_model_11)

# perform LRT
lrt_result_5 <- anova(logistic_model_4, logistic_model_5, test = "LRT")
lrt_result_5
lrt_result_6 <- anova(logistic_model_4, logistic_model_6, test = "LRT")
lrt_result_6
lrt_result_7 <- anova(logistic_model_4, logistic_model_7, test = "LRT")
lrt_result_7
lrt_result_8 <- anova(logistic_model_4, logistic_model_8, test = "LRT")
lrt_result_8
lrt_result_9 <- anova(logistic_model_4, logistic_model_9, test = "LRT")
lrt_result_9
lrt_result_10 <- anova(logistic_model_4, logistic_model_10, test = "LRT")
```

```
lrt_result_10
lrt_result_11 <- anova(logistic_model_4, logistic_model_11, test = "LRT")
lrt_result_11

# potential final model
logistic_model_12 <- glm(hadheartattack ~ generalhealth*sex +
  physicalactivities +
            hadangina + hadstroke + hadasthma + hadskincancer
              +
            hadcopd + haddepressivedisorder +
              hadkidneydisease +
            hadarthritis + haddiabetes*sex + smokerstatus +
              ecigaretteusage +
            alcoholdrinkers + bmi_category + agecategory +
              sex + raceethnicitycategory*sex,
           family = binomial(), data = data)
summary(logistic_model_12)
vif(logistic_model_12)

AIC(logistic_model_4)
AIC(logistic_model_12) # not a significant improvement in AIC, but added
  complexity
lrt_result_12 <- anova(logistic_model_4, logistic_model_12, test = "LRT")
lrt_result_12

# final model is chosen to be model 4 (model should be parsimonious)
summary(logistic_model_4)
AIC(logistic_model_4)

# GOF
predprob <- predict(logistic_model_4, type = "response")
roccurve <- roc(data$hadheartattack, predprob)
plot(roccurve, col = "red", lwd = 2, main = "ROC Curve for Logistic
  Regression Model")
auc_value <- auc(roccurve)
auc_value

# LOGISTIC ON IMPUTED DATA
# initial model
logistic_model_1_imp <- with(imputed_data_final, glm(hadheartattack ~
  generalhealth + physicalactivities +
                         hadangina + hadstroke +
                           hadasthma +
                           hadskincancer +
                         hadcopd +
                           haddepressivedisorder
                           + hadkidneydisease +
                         hadarthritis +
                           haddiabetes +
                           smokerstatus +
                           ecigaretteusage +
                                                  alcoholdrinkers +
                                                      bmi_category, family
                                                      = binomial()))
results_1 <- pool(logistic_model_1_imp)
summary(results_1)
insignificant_covariates_1 <-
    summary(results_1)[summary(results_1)$p.value > 0.05, ]
insignificant_covariates_1

AIC(logistic_model_1)
aic_model_1 <- sapply(logistic_model_1_imp$analyses, AIC)
mean_aic_model_1 <- mean(aic_model_1)
mean_aic_model_1

# adding confounders improves AIC similar to the complete case analysis
logistic_model_2_imp <- with(imputed_data_final, glm(hadheartattack ~
    generalhealth + physicalactivities +
                                                  hadangina + hadstroke +
                                                      hadasthma +
                                                      hadskincancer +
                                                  hadcopd +
                                                      haddepressivedisorder
                                                      + hadkidneydisease +
                                                  hadarthritis +
                                                      haddiabetes +
                                                      smokerstatus +
                                                      ecigaretteusage +
                                                  alcoholdrinkers +
                                                      bmi_category +
                                                      agecategory + sex +
                                                      raceethnicitycategory,
                                                family = binomial()))

results_2 <- pool(logistic_model_2_imp)
summary(results_2)
insignificant_covariates_2 <-
    summary(results_2)[summary(results_2)$p.value > 0.05, ]
insignificant_covariates_2

# check AIC
aic_model_1 <- sapply(logistic_model_1_imp$analyses, AIC)
aic_model_2 <- sapply(logistic_model_2_imp$analyses, AIC)
mean_aic_model_1 <- mean(aic_model_1)
mean_aic_model_1
mean_aic_model_2 <- mean(aic_model_2)
mean_aic_model_2
100*(mean_aic_model_1 - mean_aic_model_2)/mean_aic_model_1

# assessment of effect modifications
# interaction term: sex * raceethnicitycategory
logistic_model_3_imp <- with(imputed_data_final, glm(hadheartattack ~
   generalhealth + physicalactivities +
                                                  hadangina + hadstroke +
                                                     hadasthma +
                                                     hadskincancer +
                                                  hadcopd +
                                                     haddepressivedisorder
                                                     + hadkidneydisease +
                                                  hadarthritis +
                                                     haddiabetes +
                                                     smokerstatus +
                                                     ecigaretteusage +
                                                  alcoholdrinkers +
                                                     bmi_category +
                                                     agecategory + sex +
                                                     raceethnicitycategory*sex,
                                                family = binomial()))
results_3 <- pool(logistic_model_3_imp)
insignificant_covariates_3 <-
   summary(results_3)[summary(results_3)$p.value > 0.05, ]
insignificant_covariates_3
mean_aic_model_2
aic_model_3 <- sapply(logistic_model_3_imp$analyses, AIC)
mean_aic_model_3 <- mean(aic_model_3)
mean_aic_model_3

# interaction term: sex * agecategory
logistic_model_4_imp <- with(imputed_data_final, glm(hadheartattack ~
   generalhealth + physicalactivities +
                                                  hadangina + hadstroke +
                                                     hadasthma +
                                                     hadskincancer +
                                                  hadcopd +
                                                     haddepressivedisorder
                                                     + hadkidneydisease +
                                                  hadarthritis +
                                                     haddiabetes +
                                                     smokerstatus +
                                                     ecigaretteusage +
                                                  alcoholdrinkers +
                                                     bmi_category +
                                                     agecategory*sex +
                                                     sex +
                                                     raceethnicitycategory,
                                                family = binomial()))
results_4 <- pool(logistic_model_4_imp)
insignificant_covariates_4 <-
   summary(results_4)[summary(results_4)$p.value > 0.05, ]
insignificant_covariates_4
mean_aic_model_2
aic_model_4 <- sapply(logistic_model_4_imp$analyses, AIC)
mean_aic_model_4 <- mean(aic_model_4)
```

```
mean_aic_model_4

# interaction term: sex * bmi_category
logistic_model_5_imp <- with(imputed_data_final, glm(hadheartattack ~
   generalhealth + physicalactivities +
                                                  hadangina + hadstroke +
                                                     hadasthma +
                                                     hadskincancer +
                                                  hadcopd +
                                                     haddepressivedisorder
                                                     + hadkidneydisease +
                                                  hadarthritis +
                                                     haddiabetes +
                                                     smokerstatus +
                                                     ecigaretteusage +
                                                  alcoholdrinkers +
                                                     bmi_category*sex +
                                                     agecategory + sex +
                                                     raceethnicitycategory,
                                                family = binomial()))
results_5 <- pool(logistic_model_5_imp)
insignificant_covariates_5 <-
   summary(results_5)[summary(results_5)$p.value > 0.05, ]
insignificant_covariates_5
mean_aic_model_2
aic_model_5 <- sapply(logistic_model_5_imp$analyses, AIC)
mean_aic_model_5 <- mean(aic_model_5)
mean_aic_model_5

# interaction term: sex * smokerstatus
logistic_model_6_imp <- with(imputed_data_final, glm(hadheartattack ~
   generalhealth + physicalactivities +
                                                  hadangina + hadstroke +
                                                     hadasthma +
                                                     hadskincancer +
                                                  hadcopd +
                                                     haddepressivedisorder
                                                     + hadkidneydisease +
                                                  hadarthritis +
                                                     haddiabetes +
                                                     smokerstatus*sex +
                                                     ecigaretteusage +
                                                  alcoholdrinkers +
                                                     bmi_category +
                                                     agecategory + sex +
                                                     raceethnicitycategory,
                                                family = binomial()))
results_6 <- pool(logistic_model_6_imp)
insignificant_covariates_6 <-
   summary(results_6)[summary(results_6)$p.value > 0.05, ]
insignificant_covariates_6
```

```
mean_aic_model_2
aic_model_6 <- sapply(logistic_model_6_imp$analyses, AIC)
mean_aic_model_6 <- mean(aic_model_6)
mean_aic_model_6

# interaction term: sex * ecigaretteusage
logistic_model_7_imp <- with(imputed_data_final, glm(hadheartattack ~
   generalhealth + physicalactivities +
                                                  hadangina + hadstroke +
                                                     hadasthma +
                                                     hadskincancer +
                                                  hadcopd +
                                                     haddepressivedisorder
                                                     + hadkidneydisease +
                                                  hadarthritis +
                                                     haddiabetes +
                                                     smokerstatus +
                                                     ecigaretteusage*sex +
                                                  alcoholdrinkers +
                                                     bmi_category +
                                                     agecategory + sex +
                                                     raceethnicitycategory,
                                                family = binomial()))
results_7 <- pool(logistic_model_7_imp)
insignificant_covariates_7 <-
   summary(results_7)[summary(results_7)$p.value > 0.05, ]
insignificant_covariates_7
mean_aic_model_2
aic_model_7 <- sapply(logistic_model_7_imp$analyses, AIC)
mean_aic_model_7 <- mean(aic_model_7)
mean_aic_model_7

# interaction term: sex * haddiabetes
logistic_model_8_imp <- with(imputed_data_final, glm(hadheartattack ~
   generalhealth + physicalactivities +
                                                  hadangina + hadstroke +
                                                     hadasthma +
                                                     hadskincancer +
                                                  hadcopd +
                                                     haddepressivedisorder
                                                     + hadkidneydisease +
                                                  hadarthritis +
                                                     haddiabetes*sex +
                                                     smokerstatus +
                                                     ecigaretteusage +
                                                  alcoholdrinkers +
                                                     bmi_category +
                                                     agecategory + sex +
                                                     raceethnicitycategory,
                                                family = binomial()))
results_8 <- pool(logistic_model_8_imp)
```

```
insignificant_covariates_8 <-
   summary(results_8)[summary(results_8)$p.value > 0.05, ]
insignificant_covariates_8
mean_aic_model_2
aic_model_8 <- sapply(logistic_model_8_imp$analyses, AIC)
mean_aic_model_8 <- mean(aic_model_8)
mean_aic_model_8


# interaction term: sex * generalhealth
logistic_model_9_imp <- with(imputed_data_final, glm(hadheartattack ~
   generalhealth*sex + physicalactivities +
                                                 hadangina + hadstroke +
                                                    hadasthma +
                                                    hadskincancer +
                                                 hadcopd +
                                                    haddepressivedisorder
                                                    + hadkidneydisease +
                                                 hadarthritis +
                                                    haddiabetes +
                                                    smokerstatus +
                                                    ecigaretteusage +
                                                 alcoholdrinkers +
                                                    bmi_category +
                                                    agecategory + sex +
                                                    raceethnicitycategory,
                                               family = binomial()))
results_9 <- pool(logistic_model_9_imp)
insignificant_covariates_9 <-
   summary(results_9)[summary(results_9)$p.value > 0.05, ]
insignificant_covariates_9
mean_aic_model_2
aic_model_9 <- sapply(logistic_model_9_imp$analyses, AIC)
mean_aic_model_9 <- mean(aic_model_9)
mean_aic_model_9

# inclusion of interaction terms
logistic_model_10_imp <- with(imputed_data_final, glm(hadheartattack ~
   generalhealth*sex + physicalactivities +
                                                  hadangina + hadstroke +
                                                     hadasthma +
                                                     hadskincancer +
                                                  hadcopd +
                                                     haddepressivedisorder
                                                     + hadkidneydisease +
                                                  hadarthritis +
                                                     haddiabetes*sex +
                                                     smokerstatus +
                                                     ecigaretteusage +
                                                  alcoholdrinkers +
                                                     bmi_category +
                                                      agecategory*sex +
                                                      sex +
                                                      raceethnicitycategory*sex,
                                                 family = binomial()))
results_10 <- pool(logistic_model_10_imp)
insignificant_covariates_10 <-
    summary(results_10)[summary(results_10)$p.value > 0.05, ]
insignificant_covariates_10
mean_aic_model_2
aic_model_10 <- sapply(logistic_model_10_imp$analyses, AIC)
mean_aic_model_10 <- mean(aic_model_10)
mean_aic_model_10
100*(mean_aic_model_2 - mean_aic_model_10)/mean_aic_model_2

# second model is kept as the final because inclusion of effect
    modifications
# doesn't result in a significant improvement (checked by AIC), but
    increases the
# complexity of the model

coef_table <- summary(results_2)
OR_imputed <- exp(coef_table[, "estimate"])
lower_CI_imputed <- exp(coef_table[, "estimate"] - 1.96 * coef_table[,
    "std.error"])
upper_CI_imputed <- exp(coef_table[, "estimate"] + 1.96 * coef_table[,
    "std.error"])
OR_results_imputed <- data.frame(
  Term = coef_table$term,
  OR_Imputed = OR_imputed,
  Lower_95_CI_Imputed = lower_CI_imputed,
  Upper_95_CI_Imputed = upper_CI_imputed
)
OR_results_imputed
merged_results <- merge(OR_results_imputed, OR_results_complete, by =
    "Term")
merged_results
print(merged_results)

imputed_datasets <- complete(imputed_data_final, action = "all")

auc_values <- c()
for (i in 1:length(imputed_datasets)) {
  data_imputed <- imputed_datasets[[i]]
  data_imputed <- na.omit(data_imputed)
  model <- glm(hadheartattack ~ generalhealth + physicalactivities +
               hadangina + hadstroke + hadasthma + hadskincancer +
               hadcopd + haddepressivedisorder + hadkidneydisease +
               hadarthritis + haddiabetes + smokerstatus +
                   ecigaretteusage +
               alcoholdrinkers + bmi_category + agecategory*sex + sex +
                   raceethnicitycategory,
              family = binomial(), data = data_imputed)
  predprob <- predict(model, type = "response")
  roccurve <- roc(data_imputed$hadheartattack, predprob)
  auc_values[i] <- auc(roccurve)
}
print(auc_values)
mean_auc <- mean(auc_values)
mean_auc

# GENERALIZED ORDINAL MODEL
data$bmi_category <- factor(data$bmi_category,
                           levels = c("normalweight", "overweight",
                              "obese"),
                           ordered = TRUE)

is.ordered(data$bmi_category)
unique(data$bmi_category)

levels(data$region)
dim(data)
data_selection_ordinal <- data |> dplyr::select(-c("state", "sex",
    "hadheartattack",
                                              "agecategory", "bmi",
                                                 "raceethnicitycategory"))
colnames(data_selection_ordinal)
str(data)

nc <- detectCores() - 1
cl <- makeCluster(nc)
registerDoParallel(cl)

# full model
model_ordinal_1 <- vglm(bmi_category ~ generalhealth + physicalactivities
    +
                         sleephours + hadangina + hadstroke + hadasthma +
                         hadskincancer + hadcopd + haddepressivedisorder +
                         hadkidneydisease + hadarthritis + haddiabetes +
                         smokerstatus + ecigaretteusage + alcoholdrinkers +
                         region,
                       family = cumulative(parallel=FALSE, reverse=TRUE),
                       data = data)
model_summary_1 <- summary(model_ordinal_1)
model_summary_1
coefficients_1 <- coef(model_summary_1)
coefficients_1
non_significant_1 <- coefficients_1[coefficients_1[, "Pr(>|z|)"] > 0.05, ]
non_significant_1
visual_1 <- data.frame(Coefficient = rownames(non_significant_1),
                      Estimate = non_significant_1[, "Estimate"],
                      P_Value = non_significant_1[, "Pr(>|z|)"])
View(visual_1)
```

```
AIC(model_ordinal_1)
length(all.vars(formula(model_ordinal_1))[-1])

# final model with confounders
model_ordinal_2 <- vglm(bmi_category ~ generalhealth + physicalactivities
    +
                        sleephours + hadangina + hadstroke + hadasthma +
                        hadskincancer + hadcopd + haddepressivedisorder +
                        hadkidneydisease + hadarthritis + haddiabetes +
                        smokerstatus + ecigaretteusage + alcoholdrinkers +
                            region +
                        agecategory + sex + raceethnicitycategory,
                      family = cumulative(parallel=FALSE, reverse=TRUE),
                      data = data)

model_summary_2 <- summary(model_ordinal_2)
coefficients_2 <- coef(model_summary_2)
coefficients_2
non_significant_2 <- coefficients_2[coefficients_2[, "Pr(>|z|)"] > 0.05,
    , drop = FALSE]
non_significant_2
str(non_significant_2)
visual_2 <- data.frame(Coefficient = rownames(non_significant_2),
                     Estimate = non_significant_2[, "Estimate"],
                     P_Value = non_significant_2[, "Pr(>|z|)"])
View(visual_2)
AIC(model_ordinal_2)
length(all.vars(formula(model_ordinal_2))[-1])
100*(AIC(model_ordinal_1)-AIC(model_ordinal_2))/AIC(model_ordinal_1)

# LRT
deviance_diff <- deviance(model_ordinal_1) - deviance(model_ordinal_2)
deviance_diff
df_diff <- df.residual(model_ordinal_1) - df.residual(model_ordinal_2)
df_diff
p_value <- pchisq(deviance_diff, df = df_diff, lower.tail = FALSE)
p_value

# evaluation of the model excluding hadcopd, haddepressivedisorder,
    hadkidneydisease based on previous analysis
model_ordinal_3 <- vglm(bmi_category ~ generalhealth + physicalactivities
    +
                        sleephours + hadangina + hadstroke + hadasthma +
                        hadskincancer + hadarthritis + haddiabetes +
                        smokerstatus + ecigaretteusage + alcoholdrinkers +
                            region +
                        agecategory + sex + raceethnicitycategory,
                      family = cumulative(parallel=FALSE, reverse=TRUE),
                      data = data)

model_summary_3 <- summary(model_ordinal_3)
```

```
coefficients_3 <- coef(model_summary_3)
coefficients_3
non_significant_3 <- coefficients_3[coefficients_3[, "Pr(>|z|)"] > 0.05,
    , drop = FALSE]
non_significant_3
str(non_significant_3)
visual_3 <- data.frame(Coefficient = rownames(non_significant_3),
                     Estimate = non_significant_3[, "Estimate"],
                     P_Value = non_significant_3[, "Pr(>|z|)"])
View(visual_3)
AIC(model_ordinal_3)
length(all.vars(formula(model_ordinal_3))[-1])

# LRT
deviance_diff <- deviance(model_ordinal_3) - deviance(model_ordinal_2)
deviance_diff
df_diff <- df.residual(model_ordinal_3) - df.residual(model_ordinal_2)
df_diff
p_value <- pchisq(deviance_diff, df = df_diff, lower.tail = FALSE)
p_value

# evaluation of the model excluding only hadkidneydisease
model_ordinal_4 <- vglm(bmi_category ~ generalhealth + physicalactivities
    +
                       sleephours + hadangina + hadstroke + hadasthma +
                       hadskincancer + hadarthritis + haddiabetes +
                           hadcopd +
                       haddepressivedisorder + smokerstatus +
                           ecigaretteusage +
                       alcoholdrinkers + region + agecategory + sex +
                           raceethnicitycategory,
                     family = cumulative(parallel=FALSE, reverse=TRUE),
                     data = data)

model_summary_4 <- summary(model_ordinal_4)
coefficients_4 <- coef(model_summary_4)
coefficients_4
non_significant_4 <- coefficients_4[coefficients_4[, "Pr(>|z|)"] > 0.05,
    , drop = FALSE]
non_significant_4
str(non_significant_4)
visual_4 <- data.frame(Coefficient = rownames(non_significant_4),
                     Estimate = non_significant_4[, "Estimate"],
                     P_Value = non_significant_4[, "Pr(>|z|)"])
View(visual_4)
AIC(model_ordinal_2)
AIC(model_ordinal_4)
length(all.vars(formula(model_ordinal_4))[-1])

# LRT
deviance_diff <- deviance(model_ordinal_4) - deviance(model_ordinal_2)
```

```
deviance_diff
df_diff <- df.residual(model_ordinal_4) - df.residual(model_ordinal_2)
df_diff
p_value <- pchisq(deviance_diff, df = df_diff, lower.tail = FALSE)
p_value
```