Heart Disease Analysis

Maalolan Bharaniraj, Ngoc Khanh Vy Le, Madhuri Krishnamurthy, Subhankar Shah

Import Dataset 1 - UCI

```
hungarian <- read.csv("unprocessed-data-set/UCI/processed.cleveland.data")
switzerland <- read.csv("unprocessed-data-set/UCI/processed.va.data")
```

Import Dataset 2 - CDC

```
heart_2016_2018 <- read.csv("unprocessed-data-set/CDC/Heart_Disease_Mortality_Data_Among_US_Adults_2016

heart_2019_2021 <- read.csv("unprocessed-data-set/CDC/Heart_Disease_Mortality_Data_Among_US_Adults_2019

stoke_2019_2021 <- read.csv("unprocessed-data-set/CDC/Stroke_Mortality_Data_Among_US_Adults_2019-2021.cs
```

Import Dataset 3 - Kaggle

```
heart_2022 <- read.csv("unprocessed-data-set/Kaggle/heart_2022_no_nans.csv")
```

Cleaning UCI data set

```
# tidy data
dataLists <- list(hungarian, switzerland)
columnNames <- c("age", "sex", "cp", "trestbps", "chol", "fbs", "restecg",
                 "thalach", "exang", "oldpeak", "slope", "ca", "thal",
                 "have_heart_disease")

# Rename function
renameColumns <- function(df) {
  names(df) <- columnNames
  return(df)
}

hungarian <- renameColumns(hungarian)
switzerland <- renameColumns(switzerland)

# merge all  data frame into 1
uci <- rbind(hungarian, switzerland)

uci <- data.frame(uci)

# convert "?" into N/A
for (col_name in names(uci)) {
  uci[[col_name]][uci[[col_name]] == "?"] <- NA
}
```

```r
uci <- uci %>%
  mutate(sex = case_when(sex == 0 ~ "female",
                         sex == 1 ~ "males")) %>%
  mutate(cp = case_when(cp == 1 ~ "typical angina",
                        cp == 2 ~ "atypical angina",
                        cp == 3 ~"non-anginal pain",
                        cp == 4 ~"asymptomatic")) %>%
  mutate(fbs = case_when(fbs == "0" ~ "true",
                         fbs == "1" ~ "false")) %>%
  mutate(restecg = case_when(restecg == 0 ~ "normal",
                             restecg == 1 ~ "ST-T wave abnormality",
                             restecg == 2 ~ "left ventricular hypertrophy")) %>%
  mutate(exang = case_when(exang == 0 ~ "no",
                           exang == 1 ~ "yes")) %>%
  mutate(slope = case_when(slope == "1" ~ "upsloping",
                           slope == "2" ~ "flat",
                           slope == "3" ~ "downsloping")) %>%
  mutate(thal = case_when(thal %in% c("3.0") ~ "normal",
                          thal %in% c("6.0") ~ "fixed defect",
                          thal %in% c("7.0", "7") ~ "reversable defect")) %>%
  mutate(have_heart_disease = case_when(have_heart_disease == 0 ~ "no",
                                        have_heart_disease %in% c(1, 2, 3, 4) ~
                                          "yes"))
# mutate to numeric
uci$trestbps <- as.numeric(uci$trestbps)
uci$chol <- as.numeric(uci$chol)
uci$thalach <- as.numeric(uci$thalach)
uci$oldpeak <- as.numeric(uci$oldpeak)
uci$ca <- as.numeric(uci$ca)

# filter the NA values
for (col in names(uci)) {
  if (is.numeric(uci[[col]])) {
    mean_val <- round(mean(uci[[col]], na.rm = TRUE))
    uci[[col]][is.na(uci[[col]])] <- mean_val
  } else {
    mode_val <- names(sort(table(uci[[col]]), decreasing = TRUE))[1]
    uci[[col]][is.na(uci[[col]])] <- mode_val
  }
}

# get unique values
uniqueValues <- sapply(uci, unique)
```

Clean the CDC data set

```r
drops <- c("Georeference")
heart_2019_2021 <- heart_2019_2021[ , !(names(heart_2019_2021) %in% drops)]

cdc <- rbind(heart_2016_2018, heart_2019_2021)
cdc <- data.frame(cdc)
```

```r
# drop unnecessary columns
drops <- c("Year", "X_lon", "Y_lat", "Class", "DataSource",
           "Data_Value_Footnote_Symbol", "Data_Value_Footnote",
           "StratificationCategory1", "StratificationCategory2",
           "Data_Value_Unit", "LocationDesc", "Topic", "TopicID")
cdc <- cdc[ , !(names(cdc) %in% drops)]

# rename
colnames(cdc)[colnames(cdc) == "Data_Value"] <- "Data_Value_Per_100000_Population"

for (col in names(cdc)) {
  if (is.numeric(cdc[[col]])) {
    mean_val <- round(mean(cdc[[col]], na.rm = TRUE), 2)
    cdc[[col]][is.na(cdc[[col]])] <- mean_val
  } else {
    mode_val <- names(sort(table(cdc[[col]]), decreasing = TRUE))[1]
    cdc[[col]][is.na(cdc[[col]])] <- mode_val
  }
}

regions <- c(
  CT = 'Northeast', ME = 'Northeast', MA = 'Northeast', NH = 'Northeast',
  RI = 'Northeast', VT = 'Northeast', NJ = 'Northeast', NY = 'Northeast',
  PA = 'Northeast', IL = 'Midwest', IN = 'Midwest', MI = 'Midwest',
  OH = 'Midwest', WI = 'Midwest', IA = 'Midwest', KS = 'Midwest',
  MN = 'Midwest', MO = 'Midwest', NE = 'Midwest', ND = 'Midwest',
  SD = 'Midwest', DE = 'South', FL = 'South', GA = 'South',
  MD = 'South', NC = 'South', SC = 'South', VA = 'South',
  DC = 'South', WV = 'South', AL = 'South', KY = 'South',
  MS = 'South', TN = 'South', AR = 'South', LA = 'South',
  OK = 'South', TX = 'South', AZ = 'West', CO = 'West',
  ID = 'West', MT = 'West', NV = 'West', NM = 'West',
  UT = 'West', WY = 'West', AK = 'West', CA = 'West',
  HI = 'West', OR = 'West', WA = 'West'
)

cdc <- cdc %>%
  mutate(Region = regions[LocationAbbr])

cdc <- na.omit(cdc)
```

cleaning Kaggle data set

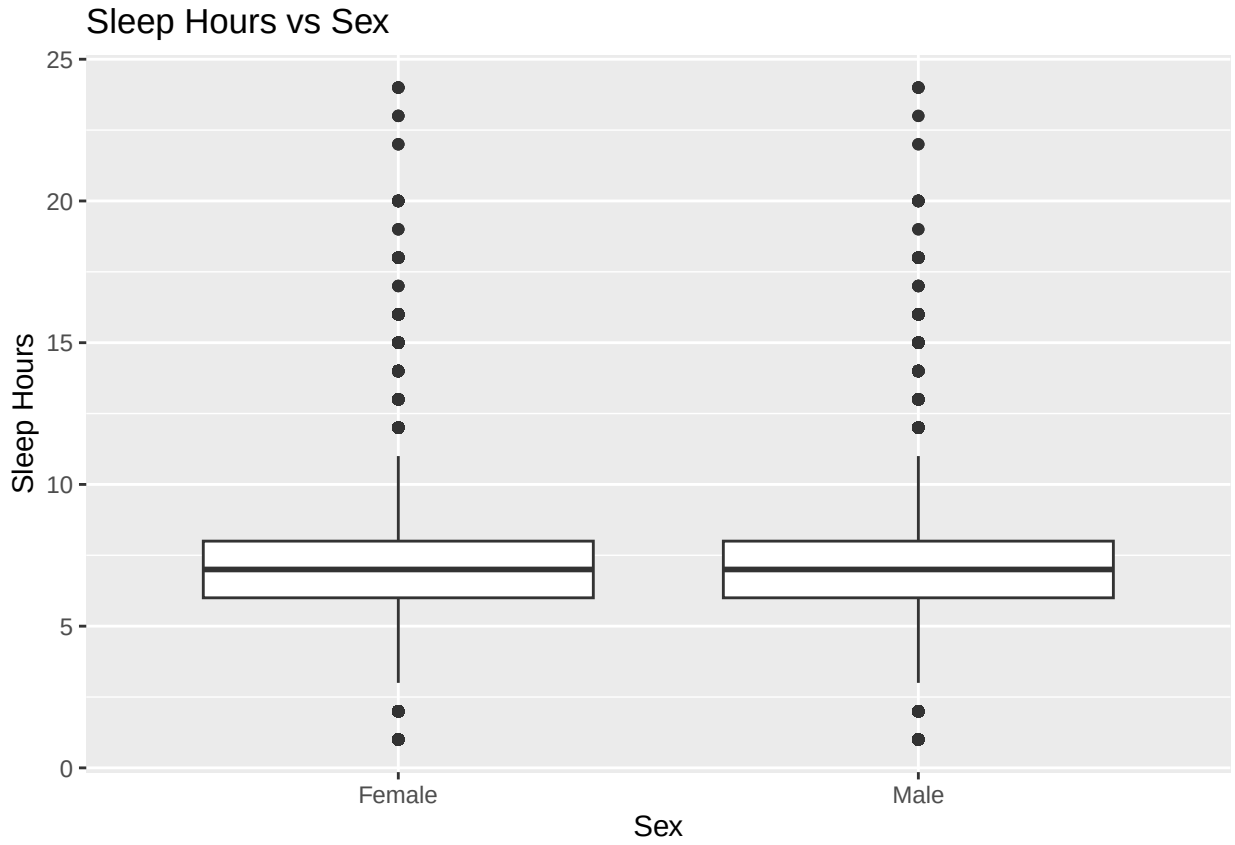
```r
# drop unnecessary columns
drops <- c("PhysicalHealthDays", "MentalHealthDays", "LastCheckupTime",
           "RemovedTeeth", "ChestScan",
           "TetanusLast10Tdap", "HighRiskLastYear",
           "StratificationCategory1", "StratificationCategory2",
           "Data_Value_Unit")
heart_2022 <- heart_2022[ , !(names(heart_2022) %in% drops)]
```

Exploratory Data Analysis

Gender vs factors to find relationships

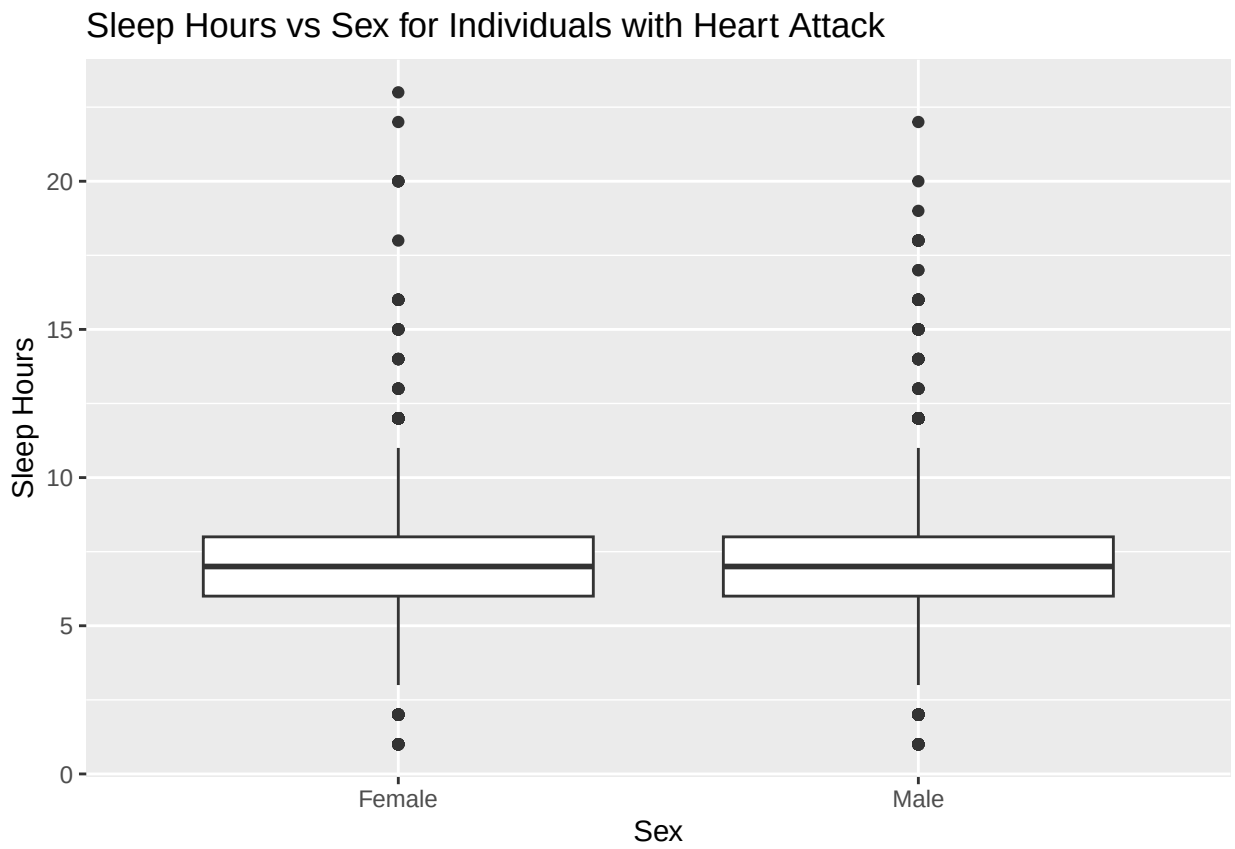
gender vs sleep and how it affect heart disease

```
ggplot(heart_2022, aes(x = Sex, y = SleepHours)) +
  geom_boxplot() +
  labs(x = "Sex", y = "Sleep Hours", title = "Sleep Hours vs Sex")
```

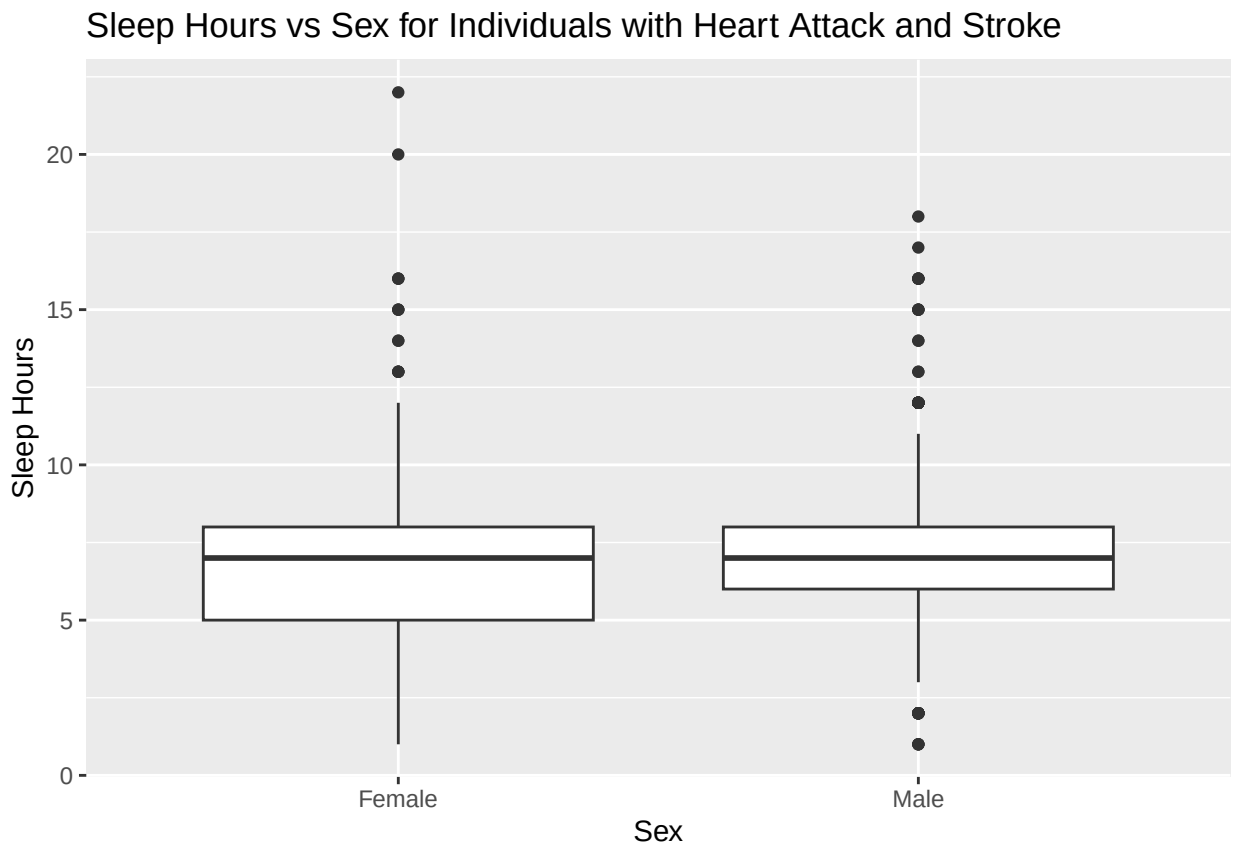


seems like male and female both have similar sleep schedule in general with females having ever so slightly more sleep than male. Let's dive in and see if it differs if we only consider the candidates with heart disease.

```
# Filter the dataset
heart_attack_data <- subset(heart_2022, HadHeartAttack == "Yes")

# Plot SleepHours vs Sex for individuals who had a heart attack
ggplot(heart_attack_data, aes(x = Sex, y = SleepHours)) +
  geom_boxplot() +
  labs(x = "Sex", y = "Sleep Hours", title = "Sleep Hours vs Sex for Individuals with Heart Attack")
```




```
heart_attack_stroke_data <- subset(heart_2022, HadHeartAttack == "Yes" & HadStroke == "Yes")

# Plot SleepHours vs Sex for individuals who had both a heart attack and a stroke
ggplot(heart_attack_stroke_data, aes(x = Sex, y = SleepHours)) +
  geom_boxplot() +
  labs(x = "Sex", y = "Sleep Hours", title = "Sleep Hours vs Sex for Individuals with Heart Attack and S
```



We can see that the sleep pattern has varied a little with slightly less sleep in the edges for females with both heart attack and stroke but on average, it hasn't varied much, so it may not really be a factor that eventually leads to heart attack or strokes ? maybe just a little. Let's explore other factors now

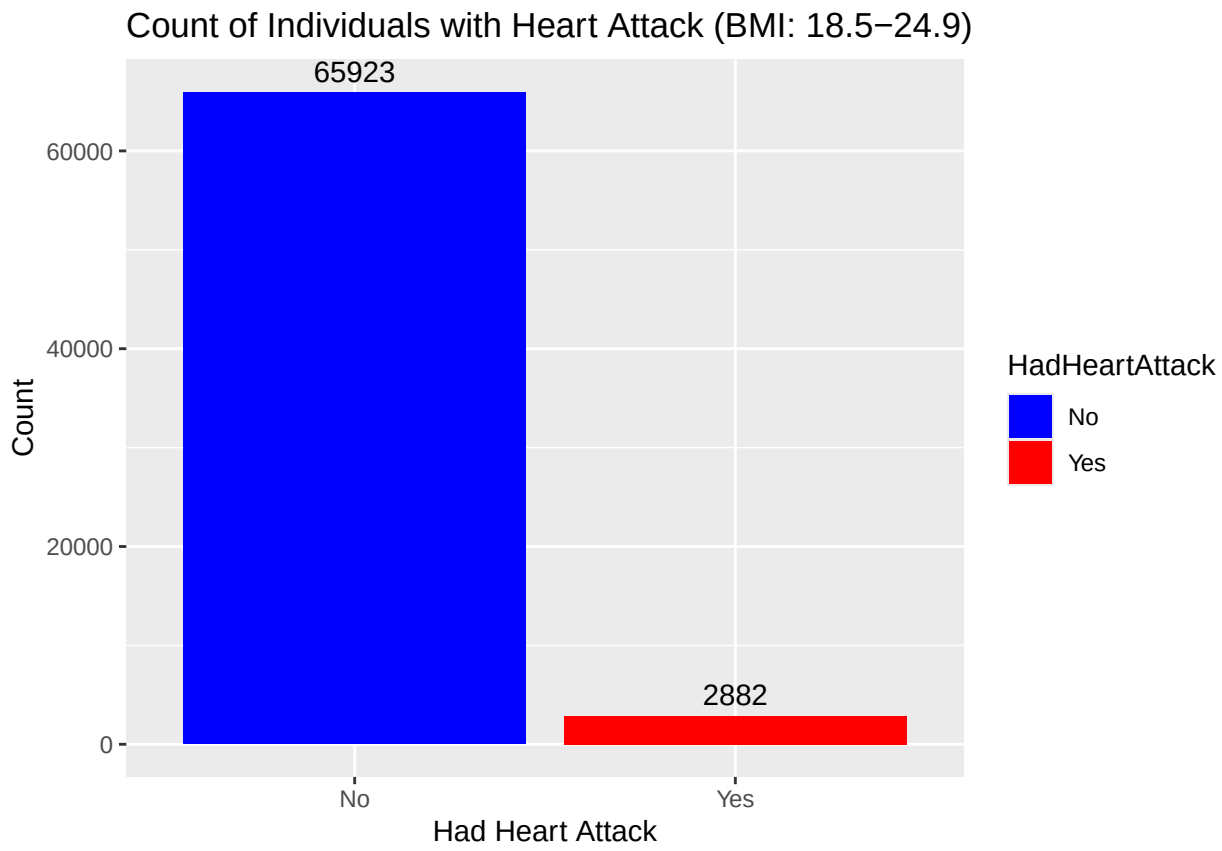
according to cdc, a healthy body mass index (BMI) for young and middle-aged adults is 18.5–24.9. so filtering out the data to just have that BMI and seeing heart attack relationships

Do people that fall under unhealthy BMI ranges likely to get heart attack ?

```
healthy_BMI <- subset(heart_2022, BMI >= 18.5 & BMI <= 24.9)
```

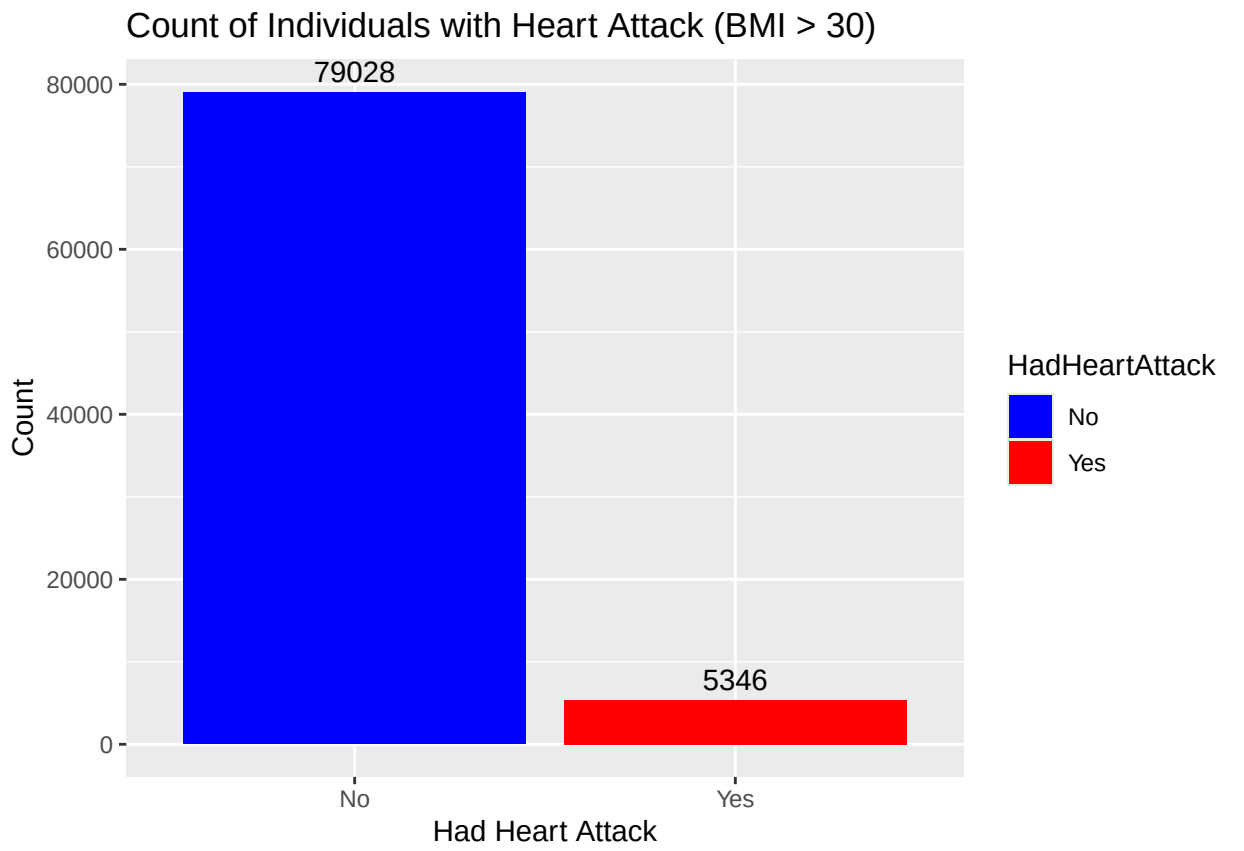
```
# Now, filter for individuals with both a heart attack and a stroke
heart_attack_stroke_filtered_data <- subset(healthy_BMI, HadHeartAttack == "Yes" & HadStroke == "Yes")

# Plot the count of individuals who had or didn't have a heart attack
ggplot(healthy_BMI, aes(x = factor(HadHeartAttack), fill = HadHeartAttack)) +
  geom_bar() +
  scale_fill_manual(values = c("Yes" = "red", "No" = "blue")) +
  labs(x = "Had Heart Attack", y = "Count", title = "Count of Individuals with Heart Attack (BMI: 18.5-1
  geom_text(stat='count', aes(label=..count..), vjust=-0.5)
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
unhealthy_BMI <- subset(heart_2022, BMI >= 30)
```

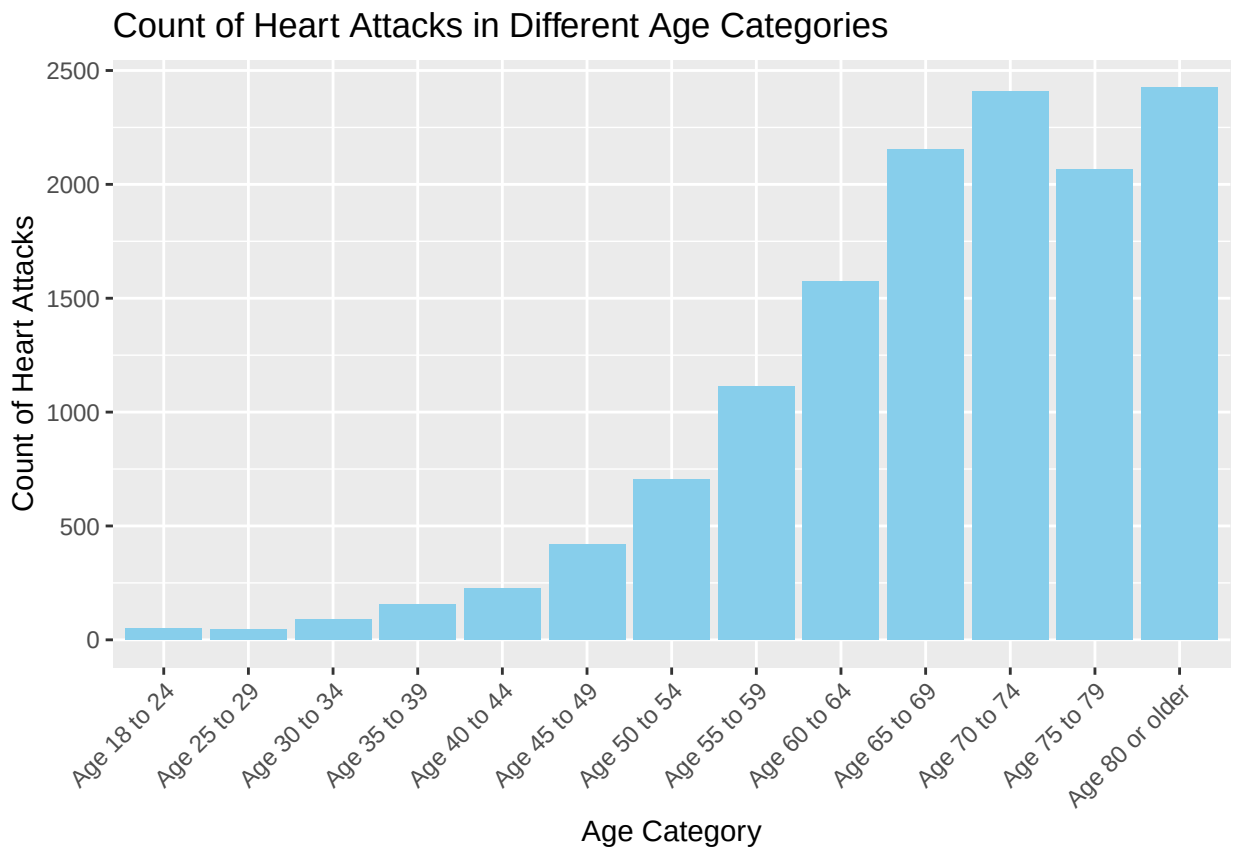
```
# Plot the count of individuals who had or didn't have a heart attack
ggplot(unhealthy_BMI, aes(x = factor(HadHeartAttack), fill = HadHeartAttack)) +
  geom_bar() +
  scale_fill_manual(values = c("Yes" = "red", "No" = "blue")) +
  labs(x = "Had Heart Attack", y = "Count", title = "Count of Individuals with Heart Attack (BMI > 30)"
  geom_text(stat='count', aes(label=..count..), vjust=-0.5)
```



as we can compare based on CDC's information, this shows us that people in unhealthy BMI range (BMI > 30) have a much higher chance of getting a heart attack. It is safe to say that this is one of the factors that lead to heart attack.

moving on, we will see if age is one of the factors for heart disease. Is older age likely to lead to heart attack ?

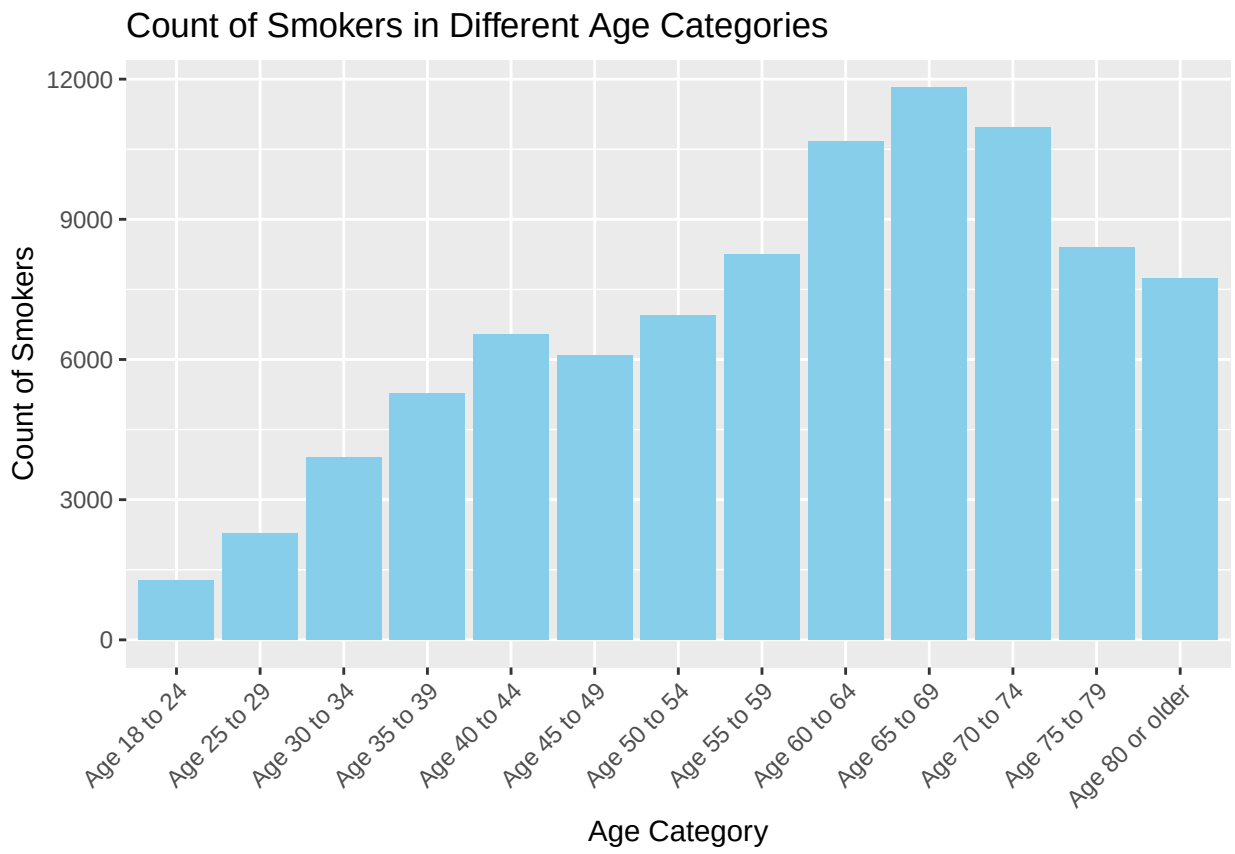
```
heart_attack_data <- subset(heart_2022, HadHeartAttack == "Yes")

heart_attack_count <- table(heart_attack_data$AgeCategory)

heart_attack_count_df <- as.data.frame(heart_attack_count)
names(heart_attack_count_df) <- c("AgeCategory", "Count")

# Plot Age vs Count using a bar graph
ggplot(heart_attack_count_df, aes(x = AgeCategory, y = Count)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(x = "Age Category", y = "Count of Heart Attacks", title = "Count of Heart Attacks in Different Ag
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for better readabili
```



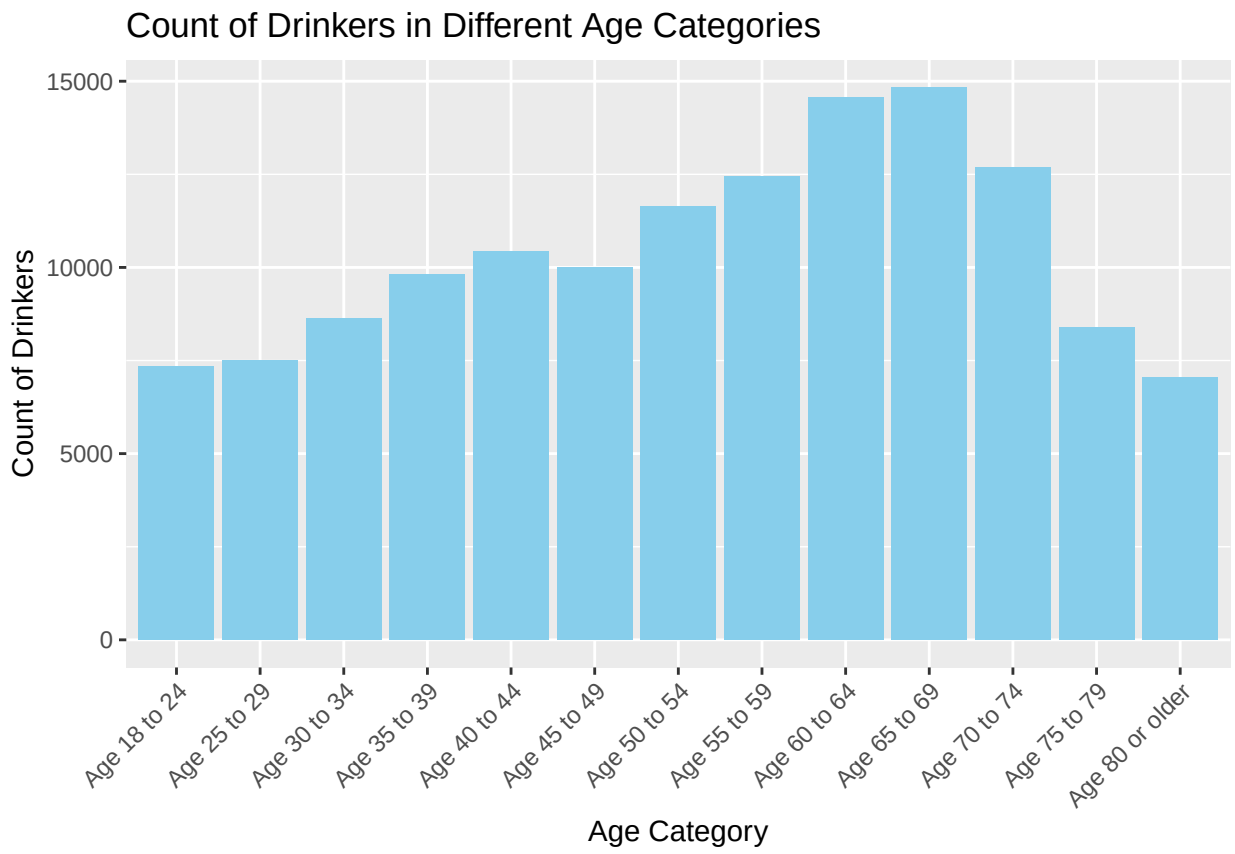
Okay, clearly we can see the trend here. There seems to be an increasing positive pattern with the number of individuals having heart attacks as the age category gets higher ie. older. To answer our question, yes older age is likely to lead to heart attack.

Let's look into what's something the highest heart attack prone age category (80 or older) is doing compared to the least heart attack prone age category. (25-29) does habits like drinking and smoking affect the likeliness of getting a heart attack ? do people who fall under the 80 or older category more likely to drink and smoke ? Let's dive in.

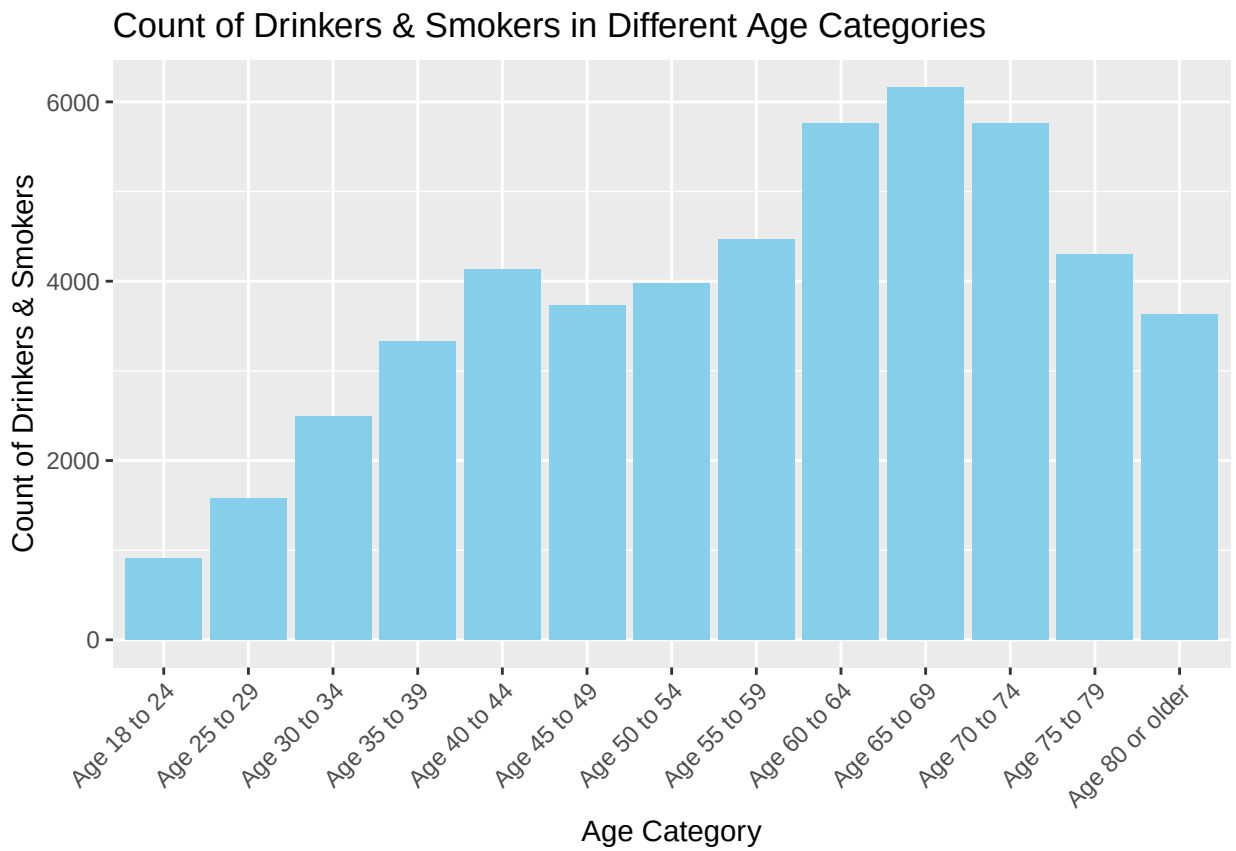
```
# Filter the dataset where SmokerStatus is either 'Current smoker - now smokes every day' or 'Former sm
habits_data <- subset(heart_2022, SmokerStatus == "Current smoker - now smokes every day" | SmokerStatus

# Group the data by AgeCategory and calculate the count in each group
smoker_count <- table(habits_data$AgeCategory)

# Convert the count to a data frame
smoker_count_df <- as.data.frame(smoker_count)
names(smoker_count_df) <- c("AgeCategory", "Count")

# Plot AgeCategory vs Count using a bar graph
ggplot(smoker_count_df, aes(x = AgeCategory, y = Count)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(x = "Age Category", y = "Count of Smokers", title = "Count of Smokers in Different Age Categories
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for better readabili
```




```r
# Filter the dataset where SmokerStatus is either 'Current smoker - now smokes every day' or 'Former sm
habits_data <- subset(heart_2022, AlcoholDrinkers == 'Yes')

# Group the data by AgeCategory and calculate the count in each group
smoker_count <- table(habits_data$AgeCategory)

# Convert the count to a data frame
smoker_count_df <- as.data.frame(smoker_count)
names(smoker_count_df) <- c("AgeCategory", "Count")

# Plot AgeCategory vs Count using a bar graph
ggplot(smoker_count_df, aes(x = AgeCategory, y = Count)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(x = "Age Category", y = "Count of Drinkers", title = "Count of Drinkers in Different Age Categor
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for better readabili
```



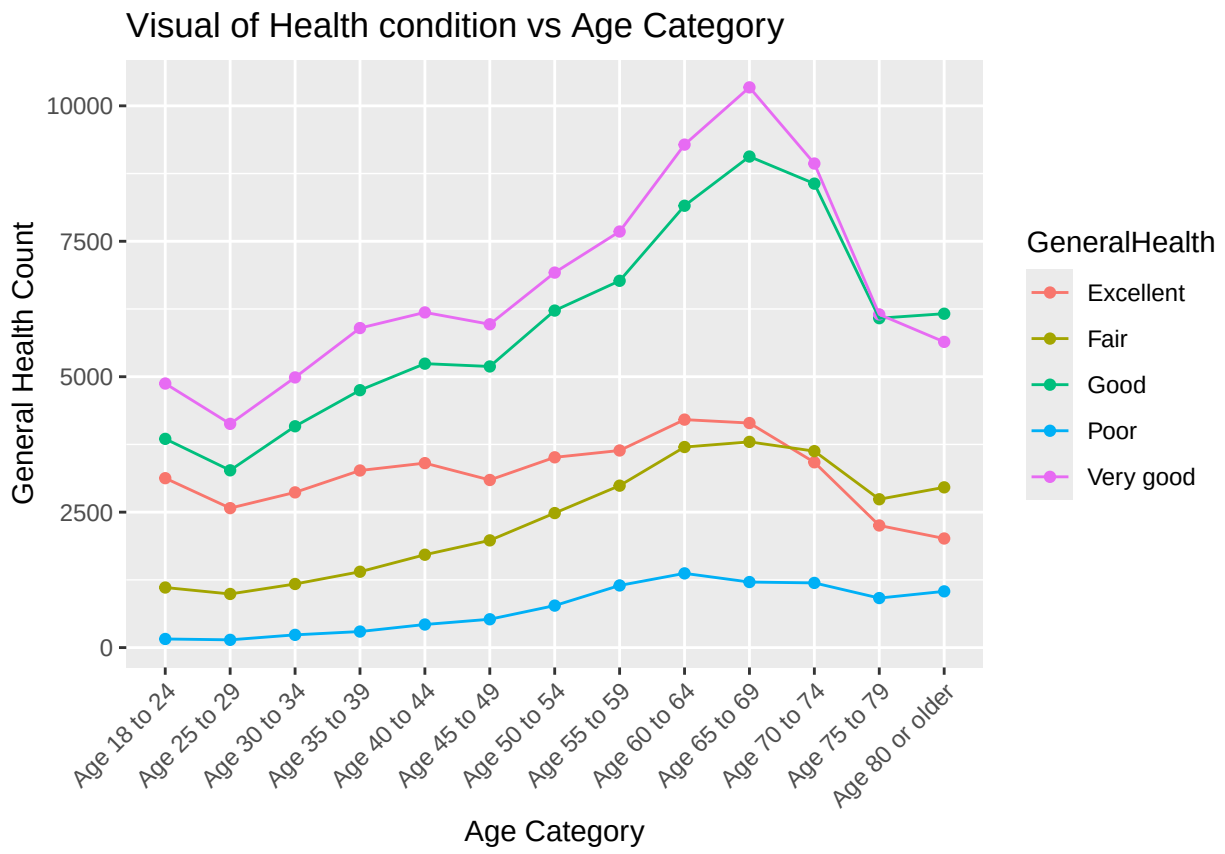

```r
# Filter the dataset where SmokerStatus is either 'Current smoker - now smokes every day' or 'Former sm
habits_data <- subset(heart_2022, AlcoholDrinkers == 'Yes' & (SmokerStatus == "Current smoker - now smok

# Group the data by AgeCategory and calculate the count in each group
smoker_count <- table(habits_data$AgeCategory)

# Convert the count to a data frame
smoker_count_df <- as.data.frame(smoker_count)
names(smoker_count_df) <- c("AgeCategory", "Count")

# Plot AgeCategory vs Count using a bar graph
ggplot(smoker_count_df, aes(x = AgeCategory, y = Count)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(x = "Age Category", y = "Count of Drinkers & Smokers", title = "Count of Drinkers & Smokers in D
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for better readabili
```



This data clearly shows us that again, people that fall under the age category of 60 and above are more likely to be smokers and drinkers in general with a peak in 65-69 and a low in 80 or older category. So, it is safe to conclude that along with older age which depreciates your healthy and body condition in general, they also seem more likely to be drinkers/smokers or both which is a key contributor to heart attacks.

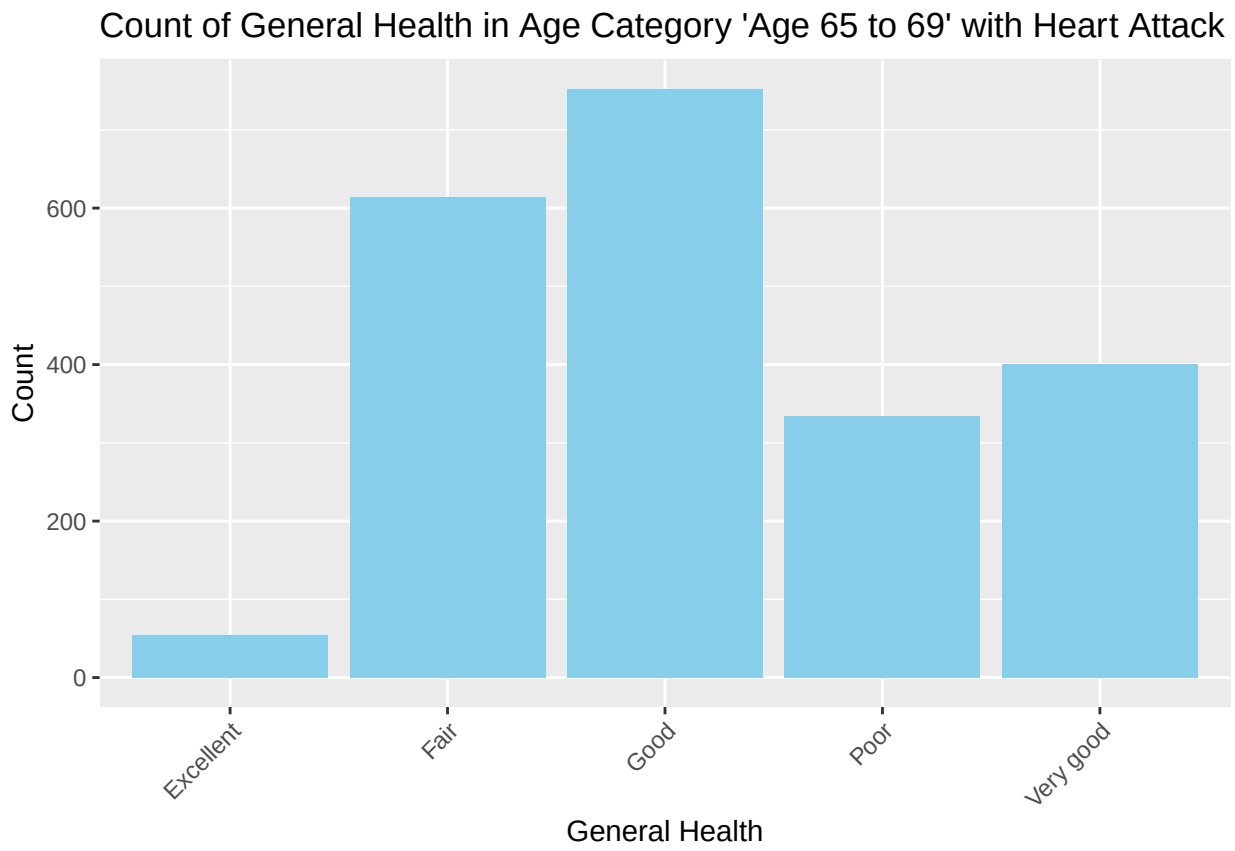
Let's see one more plot visualizing how general health changes in different age categories to conclude our EDA.

```
ggplot(heart_2022) +
  geom_point(aes(x = AgeCategory, y = ..count.., color = GeneralHealth), stat = "count") +
  geom_line(aes(x = AgeCategory, y = ..count.., group = GeneralHealth, color = GeneralHealth), stat = "
  labs(x = "Age Category", y = "General Health Count", title = "Visual of Health condition vs Age Categ
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



We can note here that generally speaking, more people with poor health condition or a fair health condition increase as the age increases too, but at the same time highest amount of people that have very good and good health condition fall under the 65-69 category. So, this raises the question, does that mean more people that are healthy in 65-69 get heart attacks or more people with poor condition get it more ?

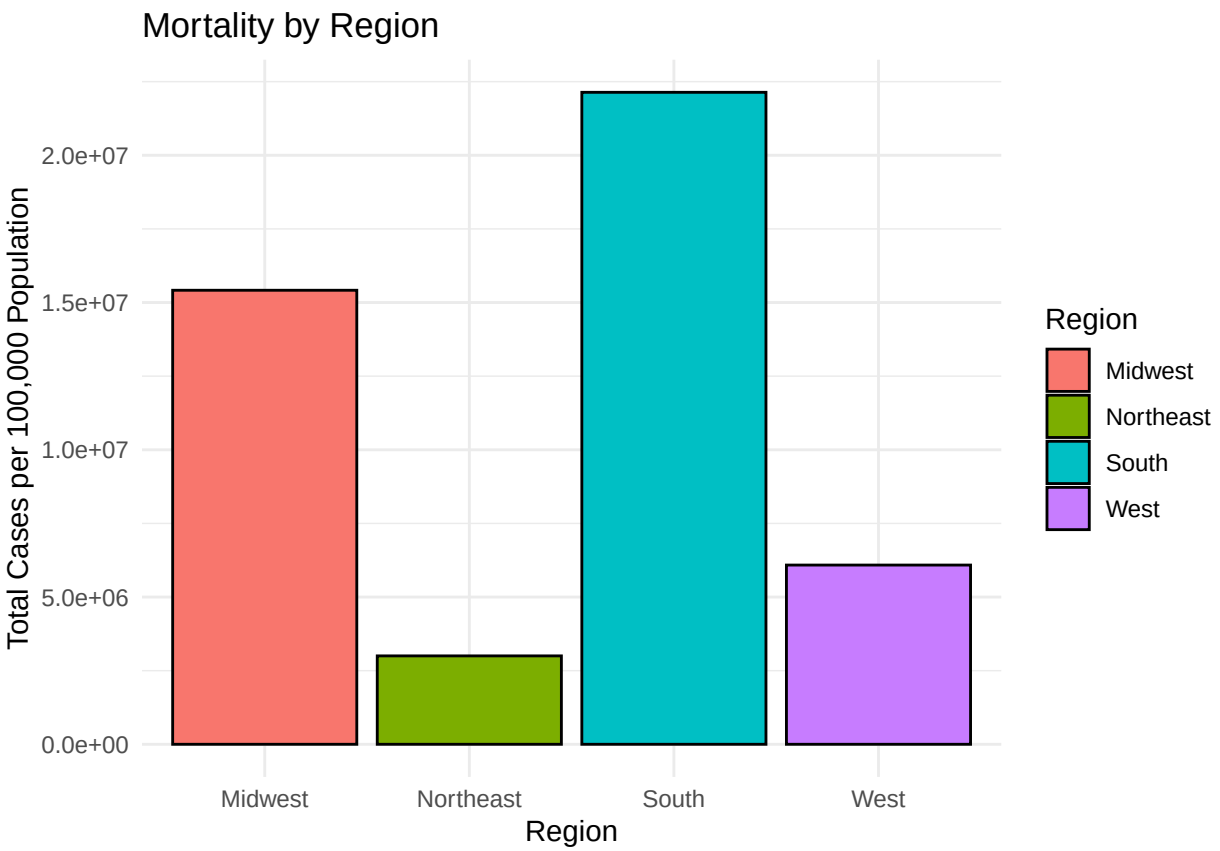
```
# Filter the dataset
age_data <- subset(heart_2022, AgeCategory == 'Age 65 to 69' & HadHeartAttack == 'Yes')

# Group the data by GeneralHealth and calculate the count in each group
age_health_count <- table(age_data$GeneralHealth)

# Convert the count to a data frame
age_health_count_df <- as.data.frame(age_health_count)
names(age_health_count_df) <- c("GeneralHealth", "Count")

# Plot GeneralHealth vs Count
ggplot(age_health_count_df, aes(x = GeneralHealth, y = Count)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(x = "General Health", y = "Count", title = "Count of General Health in Age Category 'Age 65 to 6
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for better readabili
```



Now to answer our question, people with generally good health condition get heart attacks the most in the 65-69 age category. So, from this observation we can conclude that general health probably has less to do with getting heart attacks. Comparing poor and excellent condition definitely people with poor conditions are likely to get heart attacks than ones with excellent

Identify the region

```
heart_disease_by_regions <- cdc %>%
  group_by(Region) %>%
  summarise(Total_Heart_Disease = sum(Data_Value_Per_100000_Population))

ggplot(heart_disease_by_regions, aes(x = Region,
                                     y = Total_Heart_Disease,
                                     fill = Region)) +
  geom_bar(stat = "identity", color = "black") +
  theme_minimal() +
  labs(title = "Mortality by Region",
       x = "Region",
       y = "Total Cases per 100,000 Population")
```



The South region shows a significantly higher number of cases compared to the Midwest, Northeast, and West. The South is known for a culinary tradition that often includes fried foods, higher consumption of red meat, and sweetened beverages, which can contribute to higher rates of obesity, hypertension, and diabetes - all risk factors for heart disease.

Feature Engineering

Binning Age and BMI:

Creating bins for BMI, sleep category to simplify the patterns for the machine learning algorithm.

```
# Binning BMI
bmi_bins <- c(0, 18.5, 25, 30, Inf)
bmi_labels <- c("Underweight", "Normal", "Overweight", "Obese")
heart_2022$BMICategory <- cut(heart_2022$BMI, bmi_bins,
                              labels = bmi_labels, right = FALSE)


# Define bins and labels for sleep duration categories
sleep_bins <- c(0, 5, 7, 9, Inf)
sleep_labels <- c("Less than 5 hours", "5-7 hours",
                  "7-9 hours", "More than 9 hours")

# Create a new categorical variable for sleep duration
heart_2022$SleepCategory <- cut(heart_2022$SleepHours, sleep_bins,
                                labels = sleep_labels, right = FALSE)
```

Scaling Numerical Variables:

Scaling can help improve the performance of algorithms like SVM or KNN. You can use scale() for this purpose.

```
heart_2022$ScaledHeight <- scale(heart_2022$HeightInMeters)
heart_2022$ScaledWeight <- scale(heart_2022$WeightInKilograms)
```

Feature Combination:

Creating new features based on existing ones. Creating a binary variable indicating whether a person has any pre-existing conditions.

```
heart_2022$HasConditions <- ifelse(heart_2022$HadHeartAttack == 'Yes' |
                                     heart_2022$HadAngina == 'Yes' |
                                     heart_2022$HadStroke == 'Yes' |
                                     heart_2022$HadAsthma == 'Yes' |
                                     heart_2022$HadSkinCancer == 'Yes' |
                                     heart_2022$HadCOPD == 'Yes' |
                                     heart_2022$HadCOPD == 'Yes' |
                                     heart_2022$HadDepressiveDisorder == 'Yes' |
                                     heart_2022$HadKidneyDisease == 'Yes' |
                                     heart_2022$HadArthritis == 'Yes' |
                                     heart_2022$HadDiabetes == 'Yes', 'Yes','No')

heart_2022$Vaccinated <- ifelse(heart_2022$FluVaxLast12 == 'Yes' |
                                  heart_2022$PneumoVaxEver == 'Yes', 'Yes','No')


drops <- c("FluVaxLast12", "PneumoVaxEver")
heart_2022 <- heart_2022[ , !(names(heart_2022) %in% drops)]
```

Feature Engineering for UCI data

```
# Age Binning
uci <- uci %>%
  mutate(age_group = case_when(
    age < 40 ~ "Young",
    age >= 40 & age < 60 ~ "Middle-aged",
    age >= 60 ~ "Elderly"
  ))

# Blood Pressure Scaling (Min-Max scaling)
uci$bps_scaled <- scale(uci$trestbps, center = FALSE,
                                scale = max(uci$trestbps))

# Cholesterol Binning
uci <- uci %>%
  mutate(cholesterol_group = case_when(
    chol < 200 ~ "Normal",
    chol >= 200 & chol < 240 ~ "Borderline High",
    chol >= 240 ~ "High"
  ))

# Encoding Categorical Variables
```

```r
uci <- uci %>%
  mutate(
    sex = as.factor(sex),
    fbs = as.factor(fbs),
    restecg = as.factor(restecg),
    thalach = as.factor(thalach)
  )

# One-hot encoding
one_hot_uci <- dummyVars(~ sex + fbs + restecg + thalach, data = uci) %>%
  predict(uci)
```

Modeling and Insights

Predict Heart Disease Risk

Encode categorical variables

```r
uci$sex <- as.numeric(factor(uci$sex))
uci$cp <- as.numeric(factor(uci$cp))
uci$fbs <- as.numeric(factor(uci$fbs))
uci$restecg <- as.numeric(factor(uci$restecg))
uci$exang <- as.numeric(factor(uci$exang))
uci$slope <- as.numeric(factor(uci$slope))
uci$thal <- as.numeric(factor(uci$thal))
uci$have_heart_disease <- as.factor(uci$have_heart_disease)
```

Apply logistic regression:

```r
# Combining rare levels
level_counts <- table(uci$thalach)
rare_levels <- names(level_counts[level_counts < 5])

# Replace rare levels with a common category
uci$thalach <- as.character(uci$thalach)
uci$thalach[uci$thalach %in% rare_levels] <- "Other"
uci$thalach <- factor(uci$thalach)

# Split data into training and test sets 80/20
# have_heart_disease as the dependent variable
# and all other variables as independent variables.
set.seed(1)
risk_train <- createDataPartition(uci$have_heart_disease, p=0.8, list=FALSE)
risk_train_set <- uci[risk_train,]
risk_test_set <- uci[-risk_train,]

risk_train_set$thalach <- factor(risk_train_set$thalach,
                                 levels = levels(uci$thalach))
risk_test_set$thalach <- factor(risk_test_set$thalach,
                                levels = levels(uci$thalach))

# Fit the logistic regression model
model <- glm(have_heart_disease ~ .,
             data = risk_train_set,
             family = binomial(link="logit"))
```

```r
# Summarize the model
summary(model)
```

```
##
## Call:
## glm(formula = have_heart_disease ~ ., family = binomial(link = "logit"),
##     data = risk_train_set)
##
## Coefficients: (1 not defined because of singularities)
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -1.101e+01  3.746e+00  -2.940 0.003277 **
## age                      2.559e-02  3.367e-02   0.760 0.447134
## sex                      1.855e+00  4.979e-01   3.725 0.000195 ***
## cp                      -8.159e-01  1.792e-01  -4.553 5.28e-06 ***
## trestbps                 2.906e-02  1.155e-02   2.517 0.011841 *
## chol                    -1.457e-03  3.141e-03  -0.464 0.642780
## fbs                     -2.552e-01  4.015e-01  -0.636 0.525057
## restecg                 -1.805e-01  2.607e-01  -0.692 0.488748
## thalach108               1.647e+01  2.687e+03   0.006 0.995111
## thalach110               1.870e+01  2.218e+03   0.008 0.993272
## thalach111               1.665e+00  1.968e+00   0.846 0.397422
## thalach112              -1.404e+00  1.683e+00  -0.834 0.404369
## thalach118               1.846e+01  3.662e+03   0.005 0.995979
## thalach120               6.112e-01  1.496e+00   0.409 0.682877
## thalach122              -1.138e+00  1.778e+00  -0.640 0.522003
## thalach125               2.327e+00  1.875e+00   1.241 0.214514
## thalach126               1.019e+00  1.725e+00   0.591 0.554623
## thalach128              -8.418e-01  1.566e+00  -0.537 0.590974
## thalach130               1.860e+00  2.204e+00   0.844 0.398768
## thalach131              -8.623e-01  1.795e+00  -0.480 0.631025
## thalach132               3.264e+00  1.963e+00   1.663 0.096376 .
## thalach138              -1.933e+00  2.357e+00  -0.820 0.412223
## thalach140               5.787e-01  1.497e+00   0.387 0.699055
## thalach141               1.381e+00  1.330e+00   1.039 0.298796
## thalach142               1.108e+00  1.616e+00   0.685 0.493058
## thalach143               2.970e-01  1.666e+00   0.178 0.858554
## thalach144               1.843e-01  1.663e+00   0.111 0.911745
## thalach145               6.823e-01  2.025e+00   0.337 0.736175
## thalach147              -6.356e-01  1.888e+00  -0.337 0.736454
## thalach148              -6.646e-01  1.751e+00  -0.380 0.704278
## thalach150               1.813e+00  1.682e+00   1.078 0.280922
## thalach151              -1.774e+01  2.598e+03  -0.007 0.994553
## thalach152               1.024e+00  1.552e+00   0.660 0.509333
## thalach154               5.251e-01  1.575e+00   0.333 0.738896
## thalach155               2.186e+01  2.532e+03   0.009 0.993113
## thalach156               1.414e+00  1.849e+00   0.765 0.444311
## thalach157              -3.204e-02  1.824e+00  -0.018 0.985983
## thalach158               3.571e+00  1.751e+00   2.039 0.041409 *
## thalach159               8.368e-01  1.729e+00   0.484 0.628295
## thalach160               2.098e-01  1.502e+00   0.140 0.888923
## thalach161               9.135e-02  1.608e+00   0.057 0.954701
## thalach162               1.305e+00  1.633e+00   0.799 0.424223
## thalach163               7.534e-01  1.815e+00   0.415 0.678016
```

```
## thalach165              -1.827e+00  2.507e+00  -0.729 0.466107
## thalach168              -1.236e+00  2.038e+00  -0.606 0.544210
## thalach169               1.568e+00  1.781e+00   0.881 0.378517
## thalach170              -1.963e-01  4.064e+00  -0.048 0.961475
## thalach172              -1.492e+01  2.469e+03  -0.006 0.995177
## thalach173              -1.697e+00  1.847e+00  -0.919 0.358298
## thalach174               2.413e+00  1.716e+00   1.406 0.159596
## thalach178              -1.682e+01  3.726e+03  -0.005 0.996399
## thalach179              -1.558e+01  2.822e+03  -0.006 0.995597
## thalach182               2.083e-01  1.866e+00   0.112 0.911129
## thalachOther             8.238e-01  1.262e+00   0.653 0.513726
## exang                    1.235e+00  4.122e-01   2.995 0.002742 **
## oldpeak                  6.929e-01  2.281e-01   3.038 0.002384 **
## slope                   -1.729e-01  3.583e-01  -0.483 0.629359
## ca                       1.309e+00  2.989e-01   4.379 1.19e-05 ***
## thal                     8.160e-01  3.598e-01   2.268 0.023345 *
## age_groupMiddle-aged    -3.327e-02  5.504e-01  -0.060 0.951797
## age_groupYoung           1.834e-01  1.360e+00   0.135 0.892680
## bps_scaled                      NA         NA      NA       NA
## cholesterol_groupHigh    3.652e-01  4.413e-01   0.828 0.407933
## cholesterol_groupNormal -7.820e-01  5.822e-01  -1.343 0.179217
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 548.89  on 401  degrees of freedom
## Residual deviance: 282.72  on 339  degrees of freedom
## AIC: 408.72
##
## Number of Fisher Scoring iterations: 17
```

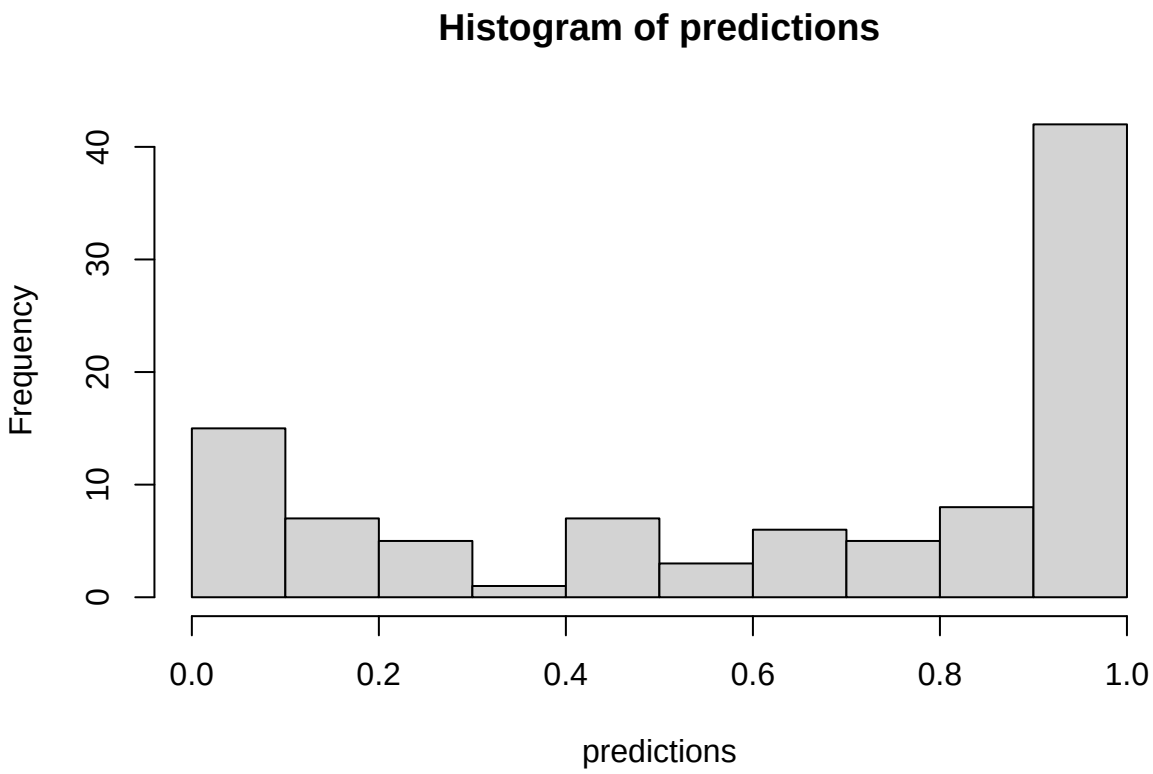
```
# calculate McFadden's R-squared for model
cat("McFadden's R-squared",
    with(summary(model), 1 - deviance/null.deviance),
         "\n")
```

```
## McFadden's R-squared 0.4849236
```

```
predictions <- predict(model, risk_test_set, type="response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```
predicted_classes <- ifelse(predictions > 0.5, "yes", "no")
predicted_classes <- factor(predicted_classes, levels = c("no", "yes"))

hist(predictions)
```




```
# Confusion matrix to see the accuracy, sensitivity, and specificity
confusionMatrix(predicted_classes, risk_test_set$have_heart_disease)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction no yes
##        no  27   8
##        yes 15  49
##
##                Accuracy : 0.7677
##                  95% CI : (0.6721, 0.8467)
##     No Information Rate : 0.5758
##     P-Value [Acc > NIR] : 5.144e-05
##
##                   Kappa : 0.5138
##
##  Mcnemar's Test P-Value : 0.2109
##
##             Sensitivity : 0.6429
##             Specificity : 0.8596
##          Pos Pred Value : 0.7714
##          Neg Pred Value : 0.7656
##              Prevalence : 0.4242
##          Detection Rate : 0.2727
##    Detection Prevalence : 0.3535
```

```
##       Balanced Accuracy : 0.7513
##
##        'Positive' Class : no
##
```

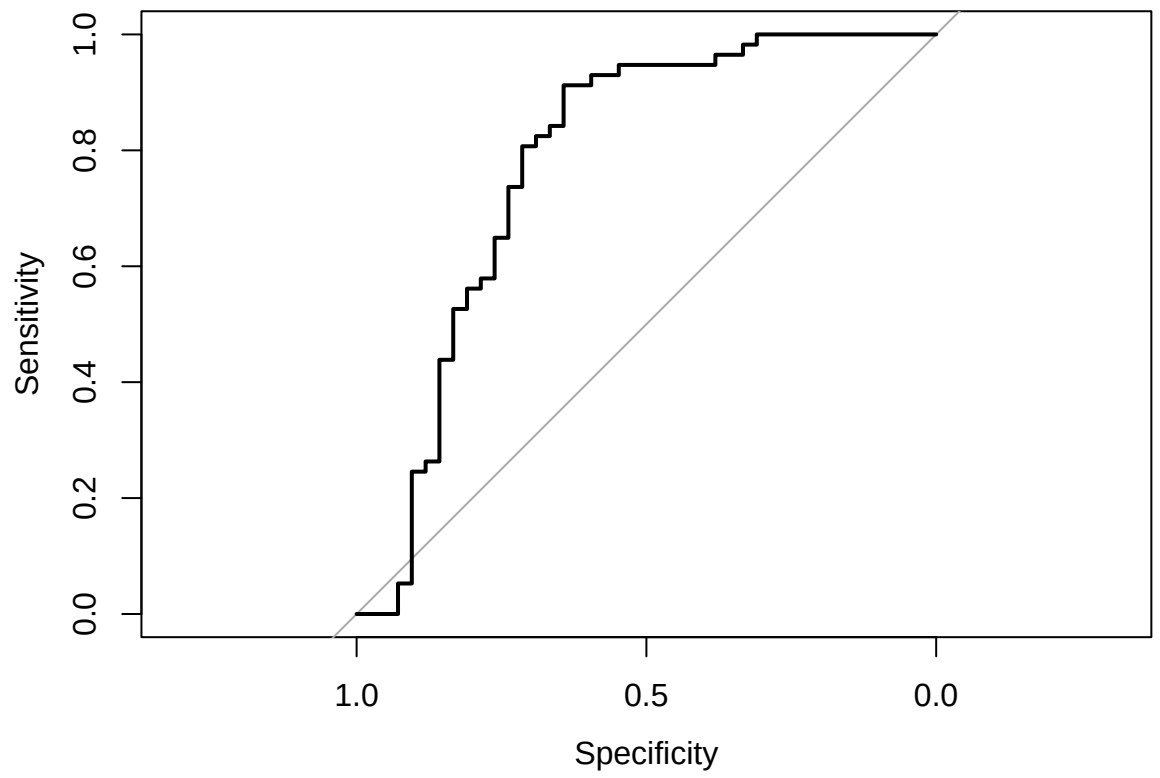
Get ROC curve

```
roc_response <- roc(response = risk_test_set$have_heart_disease,
                    predictor = as.numeric(predictions))
```

```
## Setting levels: control = no, case = yes
```

```
## Setting direction: controls < cases
```

```
plot(roc_response)
```




```
auc(roc_response)
```

```
## Area under the curve: 0.7799
```

Calculate F-1 Score:

```
P <- 49 / (49 + 15) # precision
P
```

```
## [1] 0.765625
```

```
R <- 49 / (49 + 8) # recall
R
```

```
## [1] 0.8596491
```

```r
2 * (P * R) / (P + R) # F1
```

```
## [1] 0.8099174
```

Predict Heart Acttack Risk

One-hot code the dataset

```r
heart_2022$sex <- as.numeric(factor(heart_2022$Sex))
heart_2022$GeneralHealth <- as.numeric(factor(heart_2022$GeneralHealth))
heart_2022$PhysicalActivities <- as.numeric(factor(heart_2022$PhysicalActivities))
heart_2022$HadAngina <- as.numeric(factor(heart_2022$HadAngina))
heart_2022$HadSkinCancer <- as.numeric(factor(heart_2022$HadSkinCancer))
heart_2022$HadArthritis <- as.numeric(factor(heart_2022$HadArthritis))
heart_2022$HadDiabetes <- as.numeric(factor(heart_2022$HadDiabetes))
heart_2022$HadStroke <- as.numeric(factor(heart_2022$HadStroke))
heart_2022$HadCOPD <- as.numeric(factor(heart_2022$HadCOPD))
heart_2022$HadDepressiveDisorder <- as.numeric(factor(heart_2022$HadDepressiveDisorder))
heart_2022$HadKidneyDisease <- as.numeric(factor(heart_2022$HadKidneyDisease))
heart_2022$DeafOrHardOfHearing <- as.numeric(factor(heart_2022$DeafOrHardOfHearing))
heart_2022$BlindOrVisionDifficulty <- as.numeric(factor(heart_2022$BlindOrVisionDifficulty))
heart_2022$DifficultyConcentrating <- as.numeric(factor(heart_2022$DifficultyConcentrating))
heart_2022$DifficultyWalking <- as.numeric(factor(heart_2022$DifficultyWalking))
heart_2022$DifficultyDressingBathing <- as.numeric(factor(heart_2022$DifficultyDressingBathing))
heart_2022$DifficultyErrands <- as.numeric(factor(heart_2022$DifficultyErrands))
heart_2022$SmokerStatus <- as.numeric(factor(heart_2022$SmokerStatus))
heart_2022$ECigaretteUsage <- as.numeric(factor(heart_2022$ECigaretteUsage))
heart_2022$RaceEthnicityCategory <- as.numeric(factor(heart_2022$RaceEthnicityCategory))
heart_2022$AgeCategory <- as.numeric(factor(heart_2022$AgeCategory))
heart_2022$AlcoholDrinkers <- as.numeric(factor(heart_2022$AlcoholDrinkers))
heart_2022$HIVTesting <- as.numeric(factor(heart_2022$HIVTesting))
heart_2022$CovidPos <- as.numeric(factor(heart_2022$CovidPos))
heart_2022$BMICategory <- as.numeric(factor(heart_2022$BMICategory))
heart_2022$SleepCategory <- as.numeric(factor(heart_2022$SleepCategory))
heart_2022$HasConditions <- as.numeric(factor(heart_2022$HasConditions))
heart_2022$Vaccinated <- as.numeric(factor(heart_2022$Vaccinated))
heart_2022$Sex <- as.numeric(factor(heart_2022$Sex))

heart_2022$HadHeartAttack <- as.factor(heart_2022$HadHeartAttack)
```

Apply logistic regression

```r
# Split data into training and test sets
set.seed(123) # for reproducibility
indexes <- createDataPartition(heart_2022$HadHeartAttack, p=0.8, list=FALSE)
train <- heart_2022[indexes,]
test <- heart_2022[-indexes,]

# Fit the logistic regression model
model <- glm(HadHeartAttack ~ ., data = train, family = binomial())

summary(model)
```

```
##
## Call:
## glm(formula = HadHeartAttack ~ ., family = binomial(), data = train)
```

```
##
## Coefficients: (3 not defined because of singularities)
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -4.153e+01  1.266e+02  -0.328 0.742781
## StateAlaska                 3.410e-01  1.576e-01   2.164 0.030469 *
## StateArizona                3.518e-01  1.385e-01   2.540 0.011077 *
## StateArkansas               2.851e-01  1.483e-01   1.922 0.054657 .
## StateCalifornia             2.620e-01  1.491e-01   1.758 0.078826 .
## StateColorado               2.585e-01  1.504e-01   1.719 0.085642 .
## StateConnecticut            1.221e-01  1.475e-01   0.827 0.408022
## StateDelaware               1.395e-02  1.708e-01   0.082 0.934888
## StateDistrict of Columbia -6.769e-02  2.092e-01  -0.324 0.746246
## StateFlorida                2.884e-01  1.328e-01   2.172 0.029873 *
## StateGeorgia                1.126e-01  1.433e-01   0.786 0.432084
## StateGuam                   1.051e-01  1.915e-01   0.549 0.583334
## StateHawaii                 8.142e-02  1.462e-01   0.557 0.577690
## StateIdaho                  3.780e-01  1.553e-01   2.434 0.014940 *
## StateIllinois               1.232e-01  1.748e-01   0.705 0.480716
## StateIndiana                1.626e-01  1.404e-01   1.158 0.246865
## StateIowa                   2.037e-01  1.436e-01   1.419 0.156021
## StateKansas                 1.592e-01  1.403e-01   1.135 0.256425
## StateKentucky               1.287e-01  1.617e-01   0.796 0.426081
## StateLouisiana              1.862e-01  1.548e-01   1.203 0.228826
## StateMaine                  4.056e-01  1.371e-01   2.960 0.003080 **
## StateMaryland               1.078e-01  1.336e-01   0.806 0.420044
## StateMassachusetts          2.517e-01  1.458e-01   1.726 0.084383 .
## StateMichigan               2.356e-01  1.420e-01   1.659 0.097067 .
## StateMinnesota              8.082e-02  1.368e-01   0.591 0.554619
## StateMississippi            1.197e-01  1.662e-01   0.720 0.471591
## StateMissouri               8.938e-02  1.489e-01   0.600 0.548395
## StateMontana                3.230e-01  1.475e-01   2.190 0.028501 *
## StateNebraska               3.419e-01  1.409e-01   2.427 0.015222 *
## StateNevada                 2.835e-01  1.749e-01   1.621 0.104982
## StateNew Hampshire          2.040e-01  1.493e-01   1.366 0.171927
## StateNew Jersey             5.388e-02  1.587e-01   0.339 0.734299
## StateNew Mexico             1.972e-01  1.565e-01   1.261 0.207475
## StateNew York              -1.107e-02  1.367e-01  -0.081 0.935479
## StateNorth Carolina         3.105e-01  1.672e-01   1.857 0.063300 .
## StateNorth Dakota           2.605e-01  1.651e-01   1.578 0.114496
## StateOhio                   2.420e-01  1.321e-01   1.832 0.066980 .
## StateOklahoma               1.906e-01  1.560e-01   1.222 0.221617
## StateOregon                 2.880e-01  1.622e-01   1.775 0.075859 .
## StatePennsylvania           3.196e-01  1.632e-01   1.958 0.050248 .
## StatePuerto Rico            1.979e-01  1.539e-01   1.286 0.198354
## StateRhode Island           1.955e-01  1.577e-01   1.240 0.214919
## StateSouth Carolina         5.542e-02  1.415e-01   0.392 0.695317
## StateSouth Dakota           6.084e-01  1.425e-01   4.269 1.97e-05 ***
## StateTennessee              1.193e-01  1.578e-01   0.756 0.449435
## StateTexas                  2.127e-01  1.364e-01   1.559 0.119060
## StateUtah                   2.033e-01  1.480e-01   1.374 0.169506
## StateVermont                2.672e-01  1.461e-01   1.830 0.067317 .
## StateVirgin Islands        -6.705e-02  2.938e-01  -0.228 0.819503
## StateVirginia               7.659e-02  1.417e-01   0.541 0.588826
## StateWashington             6.371e-02  1.297e-01   0.491 0.623159
```

```
## StateWest Virginia         2.838e-01  1.495e-01   1.898 0.057710 .
## StateWisconsin             1.375e-01  1.399e-01   0.983 0.325562
## StateWyoming               1.901e-01  1.659e-01   1.146 0.251750
## Sex                        7.657e-01  3.370e-02  22.718  < 2e-16 ***
## GeneralHealth             -4.349e-02  9.420e-03  -4.617 3.90e-06 ***
## PhysicalActivities        -1.005e-01  2.645e-02  -3.799 0.000145 ***
## SleepHours                 6.261e-02  1.472e-02   4.254 2.10e-05 ***
## HadAngina                  2.343e+00  2.497e-02  93.851  < 2e-16 ***
## HadStroke                  7.800e-01  3.349e-02  23.287  < 2e-16 ***
## HadAsthmaYes              -1.557e-01  3.098e-02  -5.025 5.03e-07 ***
## HadSkinCancer             -2.818e-01  3.421e-02  -8.237  < 2e-16 ***
## HadCOPD                    1.752e-01  3.189e-02   5.494 3.92e-08 ***
## HadDepressiveDisorder     -1.770e-01  3.005e-02  -5.891 3.84e-09 ***
## HadKidneyDisease           1.167e-01  3.661e-02   3.187 0.001436 **
## HadArthritis              -3.039e-01  2.482e-02 -12.243  < 2e-16 ***
## HadDiabetes                8.944e-02  1.308e-02   6.836 8.14e-12 ***
## DeafOrHardOfHearing        5.856e-02  3.168e-02   1.849 0.064515 .
## BlindOrVisionDifficulty    1.830e-01  4.048e-02   4.520 6.19e-06 ***
## DifficultyConcentrating    1.319e-01  3.547e-02   3.719 0.000200 ***
## DifficultyWalking          2.623e-01  3.004e-02   8.732  < 2e-16 ***
## DifficultyDressingBathing  5.756e-03  4.825e-02   0.119 0.905056
## DifficultyErrands          1.936e-01  3.984e-02   4.859 1.18e-06 ***
## SmokerStatus              -2.023e-01  1.292e-02 -15.659  < 2e-16 ***
## ECigaretteUsage            1.897e-02  2.113e-02   0.898 0.369317
## RaceEthnicityCategory     -5.395e-03  1.032e-02  -0.523 0.601011
## AgeCategory                1.593e-01  5.921e-03  26.908  < 2e-16 ***
## HeightInMeters            -5.382e-01  4.766e-01  -1.129 0.258808
## WeightInKilograms         -2.107e-06  4.390e-03   0.000 0.999617
## BMI                       -6.604e-03  1.275e-02  -0.518 0.604438
## AlcoholDrinkers           -2.674e-01  2.468e-02 -10.834  < 2e-16 ***
## HIVTesting                 1.239e-01  2.696e-02   4.597 4.29e-06 ***
## CovidPos                   1.765e-02  1.353e-02   1.305 0.191987
## BMICategory                7.305e-02  2.490e-02   2.933 0.003355 **
## SleepCategory             -2.345e-01  3.464e-02  -6.771 1.28e-11 ***
## ScaledHeight                      NA         NA      NA       NA
## ScaledWeight                      NA         NA      NA       NA
## HasConditions              1.728e+01  6.328e+01   0.273 0.784742
## Vaccinated                -7.954e-02  2.957e-02  -2.690 0.007142 **
## sex                               NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 83399  on 196817  degrees of freedom
## Residual deviance: 54588  on 196731  degrees of freedom
## AIC: 54762
##
## Number of Fisher Scoring iterations: 19
```

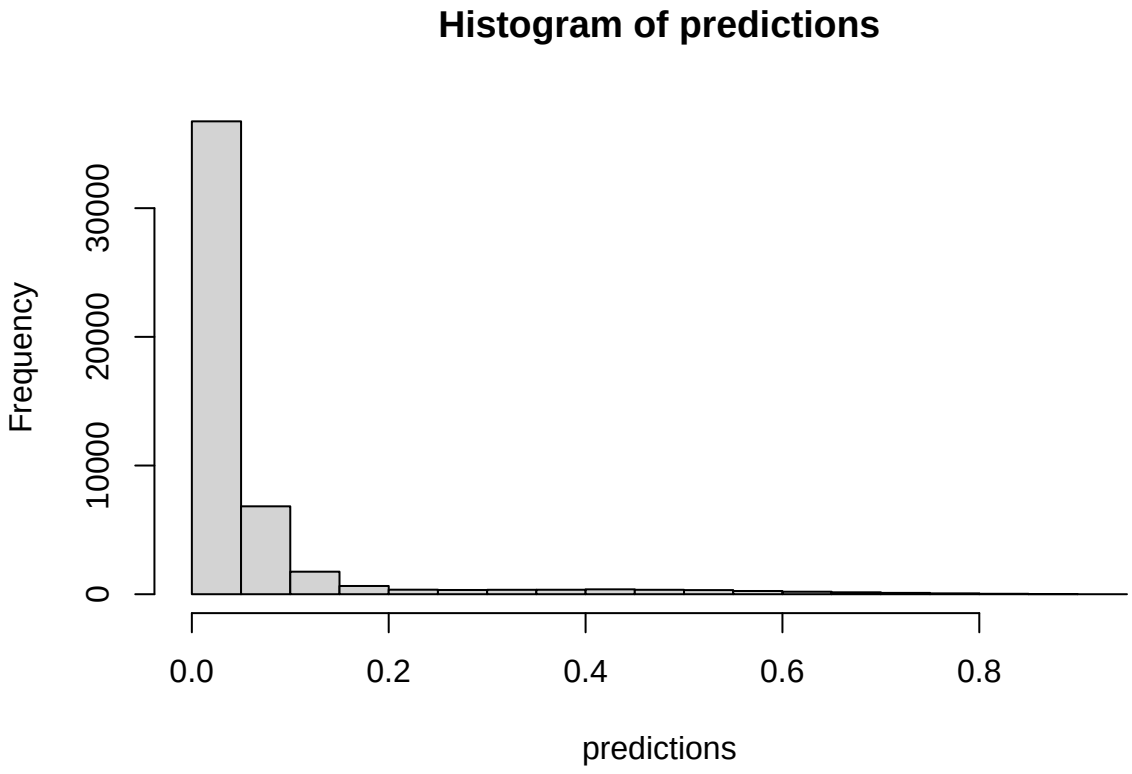
```r
# calculate McFadden's R-squared for model
cat("McFadden's R-squared",
    with(summary(model), 1 - deviance/null.deviance),
        "\n")
```

```
## McFadden's R-squared 0.3454648
```

```
predictions <- predict(model, test, type="response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```
predicted_classes <- ifelse(predictions > 0.5, "yes", "no")
predicted_classes <- factor(predicted_classes, levels = c("no", "yes"))

hist(predictions)
```




```
levels(test$HadHeartAttack) <- c("no", "yes")
levels(predicted_classes) <- c("no", "yes")

table(predicted_classes)
```

```
## predicted_classes
##    no   yes
## 48070  1134
```

```
table(test$HadHeartAttack)
```

```
##
##    no   yes
## 46517  2687
```

```
# Confusion matrix to see the accuracy, sensitivity, and specificity
confusionMatrix(predicted_classes, test$HadHeartAttack)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    no   yes
##        no  46034  2036
##        yes   483   651
##
##                Accuracy : 0.9488
##                  95% CI : (0.9468, 0.9507)
##     No Information Rate : 0.9454
##     P-Value [Acc > NIR] : 0.0003986
##
##                   Kappa : 0.3187
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9896
##             Specificity : 0.2423
##          Pos Pred Value : 0.9576
##          Neg Pred Value : 0.5741
##              Prevalence : 0.9454
##          Detection Rate : 0.9356
##    Detection Prevalence : 0.9770
##       Balanced Accuracy : 0.6159
##
##        'Positive' Class : no
##
```

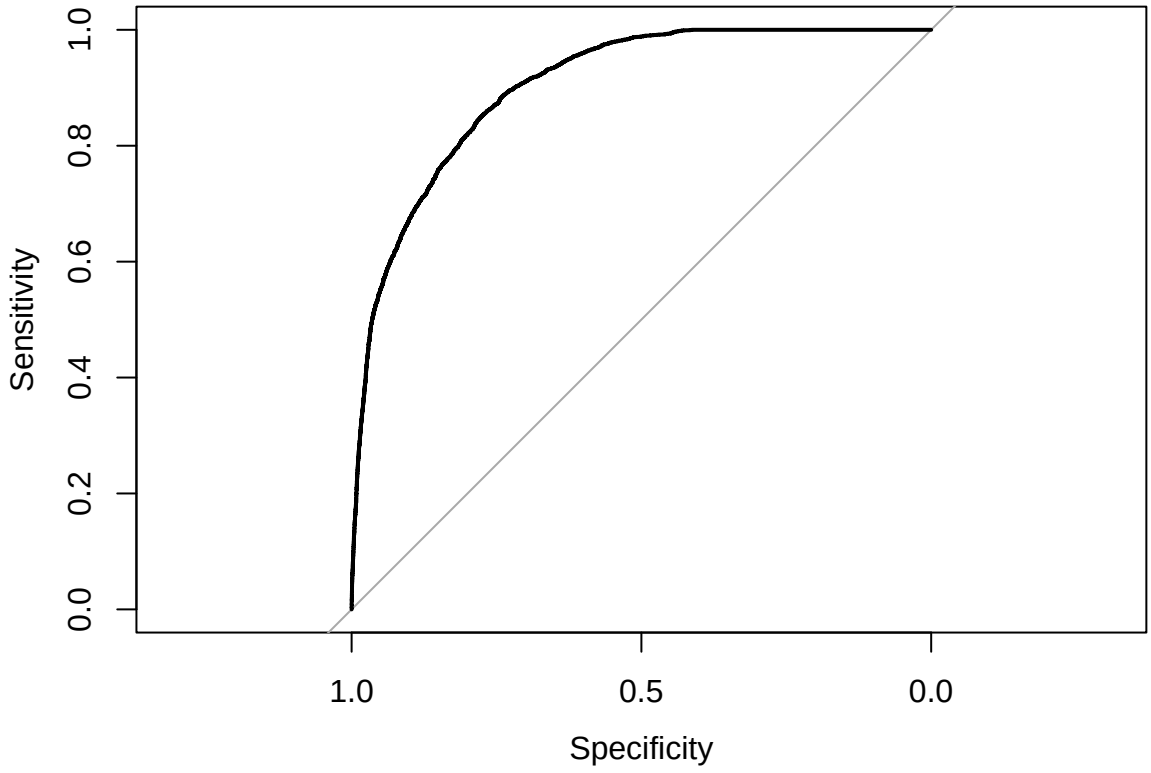
Get ROC curve

```
roc_response <- roc(response = test$HadHeartAttack, predictor = as.numeric(predictions))
```

```
## Setting levels: control = no, case = yes
```

```
## Setting direction: controls < cases
```

```
plot(roc_response)
```




```
auc(roc_response)
```

```
## Area under the curve: 0.9027
```

- As we can see the logistics regression model has the Area Under the curve of 0.9027.
- Which means that the model almost correctly predicts if the person has heart disease or not

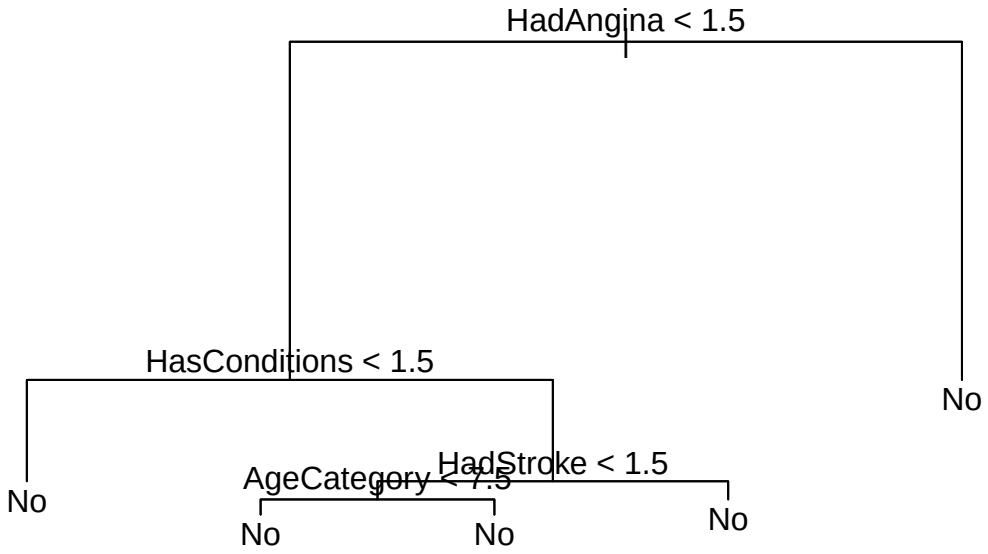
```
# Let's check the decision Tree
set.seed(2904)
tree2<-tree(HadHeartAttack~.,method="class",data=train)
```

```
## Warning in tree(HadHeartAttack ~ ., method = "class", data = train): NAs
## introduced by coercion
```

```
tree2
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 196818 83400 No ( 0.94539 0.05461 )
##    2) HadAngina < 1.5 184931 48400 No ( 0.97111 0.02889 )
##      4) HasConditions < 1.5 74115     0 No ( 1.00000 0.00000 ) *
##      5) HasConditions > 1.5 110816 42820 No ( 0.95178 0.04822 )
##       10) HadStroke < 1.5 104637 36600 No ( 0.95780 0.04220 )
##         20) AgeCategory < 7.5 38039  7382 No ( 0.98026 0.01974 ) *
##         21) AgeCategory > 7.5 66598 28380 No ( 0.94497 0.05503 ) *
##       11) HadStroke > 1.5 6179  5224 No ( 0.84998 0.15002 ) *
##    3) HadAngina > 1.5 11887 16380 No ( 0.54530 0.45470 ) *
```

```
plot(tree2)
text(tree2,pretty=0)
```




```
tree2$frame$yprob
```

```
##              No        Yes
##  [1,] 0.9453912 0.05460883
##  [2,] 0.9711081 0.02889186
##  [3,] 1.0000000 0.00000000
##  [4,] 0.9517849 0.04821506
##  [5,] 0.9577970 0.04220304
##  [6,] 0.9802571 0.01974290
##  [7,] 0.9449683 0.05503168
##  [8,] 0.8499757 0.15002428
##  [9,] 0.5453016 0.45469841
```

Let's Get the accuracy for Artificial Neural Networks

```
formula <- HadHeartAttack ~ sex + GeneralHealth + PhysicalActivities +
            HadAngina + HadSkinCancer + HadArthritis + HadDiabetes +
            HadStroke + HadCOPD + HadDepressiveDisorder + HadKidneyDisease +
            DeafOrHardOfHearing + BlindOrVisionDifficulty + DifficultyConcentrating +
            DifficultyWalking + DifficultyDressingBathing + DifficultyErrands +
            SmokerStatus + ECigaretteUsage + RaceEthnicityCategory + AgeCategory +
            AlcoholDrinkers + HIVTesting + CovidPos + BMICategory + SleepCategory +
            HasConditions + Vaccinated
```

```r
# Create the neural network model
model <- nnet(formula, data = heart_2022, size = 5, rang = 0.1, decay = 5e-4, maxit = 1000)
```

```
## # weights:  151
## initial  value 166150.068863
## final  value 52124.555183
## converged
```

Evaluate the model

```r
# Summary of the model
summary(model)
```

```
## a 28-5-1 network with 151 weights
## options were - entropy fitting  decay=5e-04
##   b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1  i9->h1
##   -0.23   -0.29   -0.60   -0.43   -0.50   -0.28   -0.35   -0.35   -0.20   -0.27
## i10->h1 i11->h1 i12->h1 i13->h1 i14->h1 i15->h1 i16->h1 i17->h1 i18->h1 i19->h1
##   -0.35   -0.36   -0.29   -0.32   -0.20   -0.35   -0.19   -0.29   -0.68   -0.26
## i20->h1 i21->h1 i22->h1 i23->h1 i24->h1 i25->h1 i26->h1 i27->h1 i28->h1
##   -0.90   -2.62   -0.23   -0.24   -0.26   -0.65   -0.60   -0.51   -0.44
##   b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2  i9->h2
##   -0.79   -1.22   -2.36   -1.15   -1.28   -0.87   -1.31   -1.36   -0.91   -0.93
## i10->h2 i11->h2 i12->h2 i13->h2 i14->h2 i15->h2 i16->h2 i17->h2 i18->h2 i19->h2
##   -0.96   -0.94   -0.87   -0.97   -1.03   -1.04   -0.95   -0.84   -2.29   -0.98
## i20->h2 i21->h2 i22->h2 i23->h2 i24->h2 i25->h2 i26->h2 i27->h2 i28->h2
##   -3.53   -8.36   -0.97   -0.95   -1.19   -2.49   -2.04   -1.71   -1.35
##   b->h3  i1->h3  i2->h3  i3->h3  i4->h3  i5->h3  i6->h3  i7->h3  i8->h3  i9->h3
##   -0.52   -0.79   -1.41   -0.78   -0.73   -0.57   -0.79   -0.90   -0.48   -0.53
## i10->h3 i11->h3 i12->h3 i13->h3 i14->h3 i15->h3 i16->h3 i17->h3 i18->h3 i19->h3
##   -0.52   -0.54   -0.52   -0.56   -0.52   -0.69   -0.49   -0.53   -1.32   -0.67
## i20->h3 i21->h3 i22->h3 i23->h3 i24->h3 i25->h3 i26->h3 i27->h3 i28->h3
##   -1.99   -5.12   -0.68   -0.61   -0.71   -1.39   -1.24   -1.00   -0.90
##   b->h4  i1->h4  i2->h4  i3->h4  i4->h4  i5->h4  i6->h4  i7->h4  i8->h4  i9->h4
##   -0.78   -1.37   -2.40   -1.21   -1.26   -0.98   -1.22   -1.48   -0.97   -1.09
## i10->h4 i11->h4 i12->h4 i13->h4 i14->h4 i15->h4 i16->h4 i17->h4 i18->h4 i19->h4
##   -0.94   -0.95   -1.05   -0.83   -0.91   -1.18   -0.93   -0.96   -2.35   -0.94
## i20->h4 i21->h4 i22->h4 i23->h4 i24->h4 i25->h4 i26->h4 i27->h4 i28->h4
##   -3.56   -8.65   -1.05   -0.96   -1.21   -2.55   -2.08   -1.66   -1.53
##   b->h5  i1->h5  i2->h5  i3->h5  i4->h5  i5->h5  i6->h5  i7->h5  i8->h5  i9->h5
##   -0.38   -0.75   -1.49   -0.69   -0.73   -0.56   -0.68   -0.86   -0.51   -0.51
## i10->h5 i11->h5 i12->h5 i13->h5 i14->h5 i15->h5 i16->h5 i17->h5 i18->h5 i19->h5
##   -0.57   -0.44   -0.58   -0.59   -0.50   -0.62   -0.44   -0.49   -1.34   -0.55
## i20->h5 i21->h5 i22->h5 i23->h5 i24->h5 i25->h5 i26->h5 i27->h5 i28->h5
##   -2.03   -5.05   -0.57   -0.61   -0.77   -1.42   -1.23   -1.01   -0.79
##  b->o h1->o h2->o h3->o h4->o h5->o
## -2.85 -2.18 -4.55 -2.66 -4.08 -2.15
```

```r
# Generate predictions on the training data
predictions <- predict(model, heart_2022, type = "class")

# Calculate accuracy
accuracy <- mean(predictions == heart_2022$HadHeartAttack)
print(paste("Accuracy of the model:", accuracy))
```

```
## [1] "Accuracy of the model: 0.945391062587899"
```

** We get an accuracy of ~94% with neural network model which is very high.

Motarlity of Heart Disease based on Region

Since there is not any columns for us to convert it into binary columns to apply logistic regression to predict the mortality rate based on region, we will use linear regression ### One-hot code

```
# Region
cdc <- cbind(cdc, model.matrix(~ Region - 1, data = cdc))

# Stratification1 and Stratification2
cdc <- cbind(cdc, model.matrix(~ Stratification1 + Stratification2 - 1, data = cdc))
```

Rename the one-hot code columns:

```
# rename the columns
colnames(cdc)[colnames(cdc) ==
                "Stratification2American Indian or Alaska Native"] <-
  "Stratification2American_Indian_or_Alaska_Native"

colnames(cdc)[colnames(cdc) ==
                "Stratification2Asian and Pacific Islander"] <-
  "Stratification2Asian_and_Pacific_Islander"

colnames(cdc)[colnames(cdc) ==
                "Stratification2More than one race"] <-
  "Stratification2More_than_one_race"

colnames(cdc)[colnames(cdc) ==
                "Stratification2Native Hawaiian or Other Pacific Islander"] <-
  "Stratification2Native_Hawaiian_or_Other_Pacific_Islander"
```

Apply linear regression

```
set.seed(2)
train_indices <- createDataPartition(cdc$Data_Value_Per_100000_Population,
                                     p = 0.8,
                                     list = FALSE)
train <- cdc[train_indices, ]
test <- cdc[-train_indices, ]

model <- lm(Data_Value_Per_100000_Population ~ RegionSouth,
            data = train)

# Summary of the model
summary(model)
```

Just South

```
##
## Call:
## lm(formula = Data_Value_Per_100000_Population ~ RegionSouth,
##     data = train)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -366.40  -19.10   12.32   14.72 2837.50
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 332.5766     0.4364  762.04   <2e-16 ***
## RegionSouth  33.8245     0.6508   51.97   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 106 on 107285 degrees of freedom
## Multiple R-squared:  0.02456,    Adjusted R-squared:  0.02455
## F-statistic:  2701 on 1 and 107285 DF,  p-value: < 2.2e-16
```

```
# Prediction
predictions <- predict(model, test)
mse <- mean((predictions - test$Data_Value_Per_100000_Population)^2)
print(paste("Mean Squared Error:", mse))
```

```
## [1] "Mean Squared Error: 10840.5873975033"
```

```
set.seed(2)
train_indices <- createDataPartition(cdc$Data_Value_Per_100000_Population,
                                     p = 0.8,
                                     list = FALSE)
train <- cdc[train_indices, ]
test <- cdc[-train_indices, ]

model <- lm(Data_Value_Per_100000_Population ~
              RegionWest + RegionSouth + RegionNortheast + RegionMidwest,
            data = train)

# Summary of the model
summary(model)
```

All regions

```
##
## Call:
## lm(formula = Data_Value_Per_100000_Population ~ RegionWest +
##     RegionSouth + RegionNortheast + RegionMidwest, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -366.40  -19.10    3.28   31.92 2837.50
##
## Coefficients: (1 not defined because of singularities)
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     344.0183     0.5572  617.43   <2e-16 ***
## RegionWest      -30.0254     1.0127  -29.65   <2e-16 ***
## RegionSouth      22.3827     0.7357   30.42   <2e-16 ***
## RegionNortheast -27.3475     1.3320  -20.53   <2e-16 ***
```

```
## RegionMidwest         NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 105.5 on 107283 degrees of freedom
## Multiple R-squared:  0.03426,    Adjusted R-squared:  0.03423
## F-statistic:  1268 on 3 and 107283 DF,  p-value: < 2.2e-16
```

```
# Prediction
predictions <- predict(model, test)
mse <- mean((predictions - test$Data_Value_Per_100000_Population)^2)
print(paste("Mean Squared Error:", mse))
```

```
## [1] "Mean Squared Error: 10734.4832670428"
```

The Multiple R-squared value of 0.03426 suggests that only about 3.43% of the variability in mortality rate is explained by the differences in regions. This is quite low, indicating that the model does not capture much of the variability in mortality rates across different regions.

The model shows statistical significance but lacks predictive strength (low R-squared and high MSE), implying that while there are some effects of region on mortality rates, there are likely many other factors not captured by this model that influence mortality rates much more significantly. It is difficult just based on this dataset to predict the heart disease mortality based on only region.

All regions + Sex + Races Try with other factors `Stratification1` and `Stratification2` (sex, races) to see if model will better. Apply linear regression and run prediction in the new model:

```
# retrain the model
set.seed(2)
train_indices <- createDataPartition(cdc$Data_Value_Per_100000_Population,
                                     p = 0.8,
                                     list = FALSE)
train <- cdc[train_indices, ]
test <- cdc[-train_indices, ]

model <- lm(Data_Value_Per_100000_Population ~
              RegionWest + RegionSouth + RegionNortheast + RegionMidwest +
              Stratification1Female + Stratification1Male +
              Stratification1Overall + Stratification2Black +
              Stratification2American_Indian_or_Alaska_Native +
              Stratification2Asian + Stratification2Hispanic +
              Stratification2Asian_and_Pacific_Islander +
              Stratification2More_than_one_race + Stratification2Overall +
              Stratification2Native_Hawaiian_or_Other_Pacific_Islander +
              Stratification2White,
            data = train)

# Summary of the model
summary(model)
```

```
##
## Call:
## lm(formula = Data_Value_Per_100000_Population ~ RegionWest +
##     RegionSouth + RegionNortheast + RegionMidwest + Stratification1Female +
##     Stratification1Male + Stratification1Overall + Stratification2Black +
##     Stratification2American_Indian_or_Alaska_Native + Stratification2Asian +
```

```
##     Stratification2Hispanic + Stratification2Asian_and_Pacific_Islander +
##     Stratification2More_than_one_race + Stratification2Overall +
##     Stratification2Native_Hawaiian_or_Other_Pacific_Islander +
##     Stratification2White, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -399.87  -52.18    3.22   49.34 2731.83
##
## Coefficients: (2 not defined because of singularities)
##                                                      Estimate Std. Error
## (Intercept)                                          342.4305     1.2034
## RegionWest                                           -30.0714     0.8901
## RegionSouth                                           22.5086     0.6466
## RegionNortheast                                      -27.3876     1.1707
## RegionMidwest                                             NA         NA
## Stratification1Female                                -27.7840     0.6930
## Stratification1Male                                   50.0737     0.6938
## Stratification1Overall                                    NA         NA
## Stratification2Black                                  57.0532     1.2992
## Stratification2American_Indian_or_Alaska_Native        1.6506     1.5036
## Stratification2Asian                                 -50.4593     1.5027
## Stratification2Hispanic                              -64.4773     1.2992
## Stratification2Asian_and_Pacific_Islander            -49.4056     1.4980
## Stratification2More_than_one_race                    -23.4514     1.4978
## Stratification2Overall                                13.0601     1.3003
## Stratification2Native_Hawaiian_or_Other_Pacific_Islander  -4.1209     1.5016
## Stratification2White                                  16.7182     1.3010
##                                                      t value Pr(>|t|)
## (Intercept)                                          284.563  < 2e-16 ***
## RegionWest                                           -33.785  < 2e-16 ***
## RegionSouth                                           34.808  < 2e-16 ***
## RegionNortheast                                      -23.393  < 2e-16 ***
## RegionMidwest                                            NA       NA
## Stratification1Female                                -40.089  < 2e-16 ***
## Stratification1Male                                   72.171  < 2e-16 ***
## Stratification1Overall                                    NA       NA
## Stratification2Black                                  43.913  < 2e-16 ***
## Stratification2American_Indian_or_Alaska_Native        1.098  0.27231
## Stratification2Asian                                 -33.579  < 2e-16 ***
## Stratification2Hispanic                              -49.630  < 2e-16 ***
## Stratification2Asian_and_Pacific_Islander            -32.980  < 2e-16 ***
## Stratification2More_than_one_race                    -15.657  < 2e-16 ***
## Stratification2Overall                                10.044  < 2e-16 ***
## Stratification2Native_Hawaiian_or_Other_Pacific_Islander  -2.744  0.00607 **
## Stratification2White                                  12.850  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 92.75 on 107272 degrees of freedom
## Multiple R-squared:  0.254,  Adjusted R-squared:  0.2539
## F-statistic:  2609 on 14 and 107272 DF,  p-value: < 2.2e-16
```

```r
# Prediction
predictions <- predict(model, test)
mse <- mean((predictions - test$Data_Value_Per_100000_Population)^2)
print(paste("Mean Squared Error:", mse))
```

```
## [1] "Mean Squared Error: 8384.09116467978"
```

The Stratification1Male and Stratification1Female suggest significant differences based on gender, with males (positive coefficient) having a higher mortality rate compared to the overall population. The positive coefficient for Stratification2Black indicates higher mortality rates for Black individuals compared to the baseline race

However the Multiple R-squared value is 0.254, which is not very high and MSE is still high. They indicate that the model still shows the lack of predictive strength. We may want to check more factors unfortunately the data is limited.