Cardiovascular Heart Disease Prediction Project

Tin Thumprateep

2025-09-11

Scope of Work

This project aims to showcase data analysis skills using R language. The dataset is synthetic and must not be cited for medical purposes.

In this dataset are:

Numerical:

- age: Age of the individual (days). (Integer)
- height: Height of the individual in centimeters. (Integer)
- weight: Weight of the individual in kilograms. (Integer)
- ap_hi: Systolic blood pressure reading. (Integer)
- ap_lo: Diastolic blood pressure reading. (Integer)
- cholesterol: Cholesterol level of the individual. (Integer)

Categorical:

- gender: Gender of the individual.(1 = female, 2 = male)
- gluc: Glucose level of the individual. (1= low, 2 = mid, 3 = high))
- smoke: Smoking status of the individual. (0 = non-smoker, 1 = smoker)
- alco: Alcohol consumption status of the individual. (0 = non-drinker, 1 = alcohol-drinker)
- active: Physical activity level of the individual. (0 = not active, 1 = active)
- cardio: Presence or absence of cardiovascular disease. (0 = without cardiovascular disease, 1 = with - cardiovascular disease)

Objectives

- Which factors poses highest risk of Cardiovascular Heart Disease?
- Which factors reduce the risk of Cardiovascular Heart Disease?
- What can we learn from this dataset?

Pre-processing

```
# Set CRAN mirror
options(repos = c(CRAN = "https://cloud.r-project.org/"))

# Install required packages if not already installed
required_packages <- c("pacman")

for(pkg in required_packages) {
  if(!require(pkg, character.only = TRUE, quietly = TRUE)) {
    install.packages(pkg)
```

```
    library(pkg, character.only = TRUE)
  }
}

set.seed(2025)
install.packages('pacman')
```

Set-up work environment:

```
##
## The downloaded binary packages are in
##  /var/folders/xx/j1c7qvvj4pq90l80z5v5b6lm0000gn/T//RtmpaGQPvf/downloaded_packages
```

```
pacman::p_load(
  rio,        # importing data
  here,       # relative file pathways
  janitor,    # data cleaning and tables
  lubridate,  # working with dates
  matchmaker, # dictionary-based cleaning
  epikit,     # age_categories() function
  tidyverse,  # data management and visualization
  skimr,      # descriptive overview of the dataframe
  corrplot,   # correlation plot
  caret,      # machine learning tools
  pROC,       # ROC curve
  car         # Multicollinearity
)

options(scipen = 999) # Remove scientific notation
```

```
heart_data = read_csv('/Users/tinthumprateep/Documents/Documents - Updated/Project/Risk Factors for Car
  janitor::clean_names()
```

Load the dataset:

```
## Rows: 70000 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (14): index, id, age, gender, height, weight, ap_hi, ap_lo, cholesterol,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
str(heart_data)
```

Explore the dataset:

```
## spc_tbl_ [70,000 x 14] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ index      : num [1:70000] 0 1 2 3 4 5 6 7 8 9 ...
##  $ id         : num [1:70000] 0 1 2 3 4 8 9 12 13 14 ...
##  $ age        : num [1:70000] 18393 20228 18857 17623 17474 ...
##  $ gender     : num [1:70000] 2 1 1 2 1 1 1 2 1 1 ...
##  $ height     : num [1:70000] 168 156 165 169 156 151 157 178 158 164 ...
##  $ weight     : num [1:70000] 62 85 64 82 56 67 93 95 71 68 ...
```

```
##  $ ap_hi      : num [1:70000] 110 140 130 150 100 120 130 130 110 110 ...
##  $ ap_lo      : num [1:70000] 80 90 70 100 60 80 80 90 70 60 ...
##  $ cholesterol: num [1:70000] 1 3 3 1 1 2 3 3 1 1 ...
##  $ gluc       : num [1:70000] 1 1 1 1 1 2 1 3 1 1 ...
##  $ smoke      : num [1:70000] 0 0 0 0 0 0 0 0 0 0 ...
##  $ alco       : num [1:70000] 0 0 0 0 0 0 0 0 0 0 ...
##  $ active     : num [1:70000] 1 1 0 1 0 0 1 1 1 0 ...
##  $ cardio     : num [1:70000] 0 1 1 1 0 0 0 1 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   index = col_double(),
##   ..   id = col_double(),
##   ..   age = col_double(),
##   ..   gender = col_double(),
##   ..   height = col_double(),
##   ..   weight = col_double(),
##   ..   ap_hi = col_double(),
##   ..   ap_lo = col_double(),
##   ..   cholesterol = col_double(),
##   ..   gluc = col_double(),
##   ..   smoke = col_double(),
##   ..   alco = col_double(),
##   ..   active = col_double(),
##   ..   cardio = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```r
summary(heart_data)
```

```
##      index             id             age            gender         height
##  Min.   :    0   Min.   :    0   Min.   :10798   Min.   :1.00   Min.   : 55.0
##  1st Qu.:17500   1st Qu.:25007   1st Qu.:17664   1st Qu.:1.00   1st Qu.:159.0
##  Median :35000   Median :50002   Median :19703   Median :1.00   Median :165.0
##  Mean   :35000   Mean   :49972   Mean   :19469   Mean   :1.35   Mean   :164.4
##  3rd Qu.:52499   3rd Qu.:74889   3rd Qu.:21327   3rd Qu.:2.00   3rd Qu.:170.0
##  Max.   :69999   Max.   :99999   Max.   :23713   Max.   :2.00   Max.   :250.0
##      weight           ap_hi             ap_lo           cholesterol
##  Min.   : 10.00   Min.   : -150.0   Min.   :  -70.00   Min.   :1.000
##  1st Qu.: 65.00   1st Qu.:  120.0   1st Qu.:   80.00   1st Qu.:1.000
##  Median : 72.00   Median :  120.0   Median :   80.00   Median :1.000
##  Mean   : 74.21   Mean   :  128.8   Mean   :   96.63   Mean   :1.367
##  3rd Qu.: 82.00   3rd Qu.:  140.0   3rd Qu.:   90.00   3rd Qu.:2.000
##  Max.   :200.00   Max.   :16020.0   Max.   :11000.00   Max.   :3.000
##       gluc           smoke              alco              active
##  Min.   :1.000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:1.000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:1.0000
##  Median :1.000   Median :0.00000   Median :0.00000   Median :1.0000
##  Mean   :1.226   Mean   :0.08813   Mean   :0.05377   Mean   :0.8037
##  3rd Qu.:1.000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:1.0000
##  Max.   :3.000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000
##      cardio
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.4997
```

```
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

```
head(heart_data)
```

```
## # A tibble: 6 x 14
##   index    id   age gender height weight ap_hi ap_lo cholesterol  gluc smoke
##   <dbl> <dbl> <dbl>  <dbl>  <dbl>  <dbl> <dbl> <dbl>       <dbl> <dbl> <dbl>
## 1     0     0 18393      2    168     62   110    80           1     1     0
## 2     1     1 20228      1    156     85   140    90           3     1     0
## 3     2     2 18857      1    165     64   130    70           3     1     0
## 4     3     3 17623      2    169     82   150   100           1     1     0
## 5     4     4 17474      1    156     56   100    60           1     1     0
## 6     5     8 21914      1    151     67   120    80           2     2     0
## # i 3 more variables: alco <dbl>, active <dbl>, cardio <dbl>
```

There could be more nonsensical values in this dataset based on the statistical summary. If we examine closely on the ap_hi and ap_lo columns the minimum values are negative, which is impossible as the human blood pressure cannot be negative and there are also absurdly high blood pressure value as well. We will remove these values later on as we process the data.

Cleaning Process

Handle Duplicate and missing values   Duplicate detection

```
# 1. check for duplicate (should return false if none were found)
any(duplicated(heart_data$id))
```

```
## [1] FALSE
```

```
# 2. Remove duplicates
heart_data_cleaned <- heart_data %>%
  distinct(id, .keep_all = TRUE)
print("Duplicates were removed")
```

```
## [1] "Duplicates were removed"
```

```
# 3. Verify
any(duplicated(heart_data_cleaned$id))
```

```
## [1] FALSE
```

Null values detection

```
# 1. Detect number of null values
if(any(is.na(heart_data_cleaned)) == FALSE) {
  print("No missing values were found")
} else {
  print("Missing values spotted")
}
```

```
## [1] "No missing values were found"
```

Data Constraints   Data Type Constraints

Let's take a look at each column mode and class before we dive in.

```
print("Mode")
```

```
## [1] "Mode"
```

```r
sapply(heart_data_cleaned, mode)
```

```
##       index          id         age      gender      height      weight
##   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"
##       ap_hi       ap_lo cholesterol        gluc       smoke        alco
##   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"
##      active      cardio
##   "numeric"   "numeric"
```

```r
print("Class")
```

```
## [1] "Class"
```

```r
sapply(heart_data_cleaned, class)
```

```
##       index          id         age      gender      height      weight
##   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"
##       ap_hi       ap_lo cholesterol        gluc       smoke        alco
##   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"
##      active      cardio
##   "numeric"   "numeric"
```

If we can recall there are certain columns which are categorical, we have to format those column accordingly.

```r
# Identify the categorical columns from dataset
categorical_columns <- c("gender", "cholesterol", "gluc", "smoke", "alco", "active", "cardio")

# Convert the columns to categorical
heart_data_cleaned <- heart_data_cleaned %>%
  mutate(across(all_of(categorical_columns), as.factor))

# Verify conversion
str(heart_data_cleaned)
```

```
## tibble [70,000 x 14] (S3: tbl_df/tbl/data.frame)
##  $ index      : num [1:70000] 0 1 2 3 4 5 6 7 8 9 ...
##  $ id         : num [1:70000] 0 1 2 3 4 8 9 12 13 14 ...
##  $ age        : num [1:70000] 18393 20228 18857 17623 17474 ...
##  $ gender     : Factor w/ 2 levels "1","2": 2 1 1 2 1 1 1 2 1 1 ...
##  $ height     : num [1:70000] 168 156 165 169 156 151 157 178 158 164 ...
##  $ weight     : num [1:70000] 62 85 64 82 56 67 93 95 71 68 ...
##  $ ap_hi      : num [1:70000] 110 140 130 150 100 120 130 130 110 110 ...
##  $ ap_lo      : num [1:70000] 80 90 70 100 60 80 80 90 70 60 ...
##  $ cholesterol: Factor w/ 3 levels "1","2","3": 1 3 3 1 1 2 3 3 1 1 ...
##  $ gluc       : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 2 1 3 1 1 ...
##  $ smoke      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ alco       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ active     : Factor w/ 2 levels "0","1": 2 2 1 2 1 1 2 2 2 1 ...
##  $ cardio     : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 2 1 1 ...
```

Data Range Constraints

Let's make nonsensical data sensical! First let's look at the statistical summary of each column to get. glimpse of what we should know before transforming our data.

```r
summary(heart_data_cleaned)
```

```
##      index             id              age           gender        height
```

```
##  Min.   :    0   Min.   :    0   Min.   :10798   1:45530   Min.   : 55.0
##  1st Qu.:17500   1st Qu.:25007   1st Qu.:17664   2:24470   1st Qu.:159.0
##  Median :35000   Median :50002   Median :19703             Median :165.0
##  Mean   :35000   Mean   :49972   Mean   :19469             Mean   :164.4
##  3rd Qu.:52499   3rd Qu.:74889   3rd Qu.:21327             3rd Qu.:170.0
##  Max.   :69999   Max.   :99999   Max.   :23713             Max.   :250.0
##      weight            ap_hi             ap_lo          cholesterol gluc
##  Min.   : 10.00   Min.   : -150.0   Min.   :  -70.00   1:52385     1:59479
##  1st Qu.: 65.00   1st Qu.:  120.0   1st Qu.:   80.00   2: 9549     2: 5190
##  Median : 72.00   Median :  120.0   Median :   80.00   3: 8066     3: 5331
##  Mean   : 74.21   Mean   :  128.8   Mean   :   96.63
##  3rd Qu.: 82.00   3rd Qu.:  140.0   3rd Qu.:   90.00
##  Max.   :200.00   Max.   :16020.0   Max.   :11000.00
##  smoke     alco      active    cardio
##  0:63831   0:66236   0:13739   0:35021
##  1: 6169   1: 3764   1:56261   1:34979
##
##
##
##
```

ap_hi and ap_lo

If we look at the ap_hi and ap_lo which represent Systolic blood pressure and Diastolic blood pressure respectively. The maximum and minimum numbers of these two columns do not make sense as they are absurdly high or negative. We have to specify reasonable range and remove any outlier.

First let's look at the context of the blood pressure. A study published that the highest blood pressure ever recorded was 370/360 mmHg, which was captured after an athlete performs sets of 85% and 100% maximum capacity.

There are certain things we need to point out about the blood pressure:

- 1. The Systolic blood pressure is always higher than Diastolic blood pressure.
- 2. The values of blood pressure cannot be negative.
- 3. Any absurdly high value would likely be an outlier.

```
# 1. Let's filter one by one. Firstly, let's filter out any rows that has Systolic blood pressure lower
heart_data_cleaned <- heart_data_cleaned %>%
  filter(ap_hi > ap_lo)

#2. Filter out the negative values.
heart_data_cleaned <- heart_data_cleaned %>%
  filter(ap_hi > 0 & ap_lo > 0)

#3. Lastly, filter out absurdly high values.
heart_data_cleaned <- heart_data_cleaned %>%
  filter(ap_hi < 370 & ap_lo < 360)

#4. Take a look at the summary statistics.
summary(heart_data_cleaned)
```

```
##      index             id             age         gender       height
##  Min.   :    0   Min.   :    0   Min.   :10798   1:44757   Min.   : 55.0
##  1st Qu.:17498   1st Qu.:25003   1st Qu.:17657   2:23952   1st Qu.:159.0
##  Median :35008   Median :50012   Median :19701             Median :165.0
##  Mean   :35000   Mean   :49974   Mean   :19464             Mean   :164.4
```

```
##  3rd Qu.:52485   3rd Qu.:74867   3rd Qu.:21324             3rd Qu.:170.0
##  Max.   :69999   Max.   :99999   Max.   :23713             Max.   :250.0
##      weight           ap_hi           ap_lo        cholesterol gluc
##  Min.   : 11.00   Min.   : 16.0   Min.   :  1.00   1:51529     1:58423
##  1st Qu.: 65.00   1st Qu.:120.0   1st Qu.: 80.00   2: 9305     2: 5069
##  Median : 72.00   Median :120.0   Median : 80.00   3: 7875     3: 5217
##  Mean   : 74.12   Mean   :126.7   Mean   : 81.27
##  3rd Qu.: 82.00   3rd Qu.:140.0   3rd Qu.: 90.00
##  Max.   :200.00   Max.   :240.0   Max.   :182.00
##  smoke     alco      active    cardio
##  0:62668   0:65044   0:13509   0:34718
##  1: 6041   1: 3665   1:55200   1:33991
##
##
##
##
```

Height and Weight

The weight and height correlation does not make any sense here as the minimum weight is 11, which is humanly impossible unless it's a children or people with extreme conditions. Since the minimum age is 30 years old, we can safely rule out children from the consideration. However, people with dwarfism can have significantly lower height and weight despite an old age. We keep this possibility in the analysis to ensure inclusivity and data integrity.

So, to resolve this issue we can setup a reasonable range of BMI and any value outside the range is filtered out.

```r
heart_data_cleaned <- heart_data_cleaned %>%
  mutate(
    BMI = weight / ((height/100)^2),
    flag = case_when(
      height < 100 | height > 250 ~ "impossible_height",
      weight < 25  | weight > 300 ~ "impossible_weight",
      BMI < 10 | BMI > 80 ~ "impossible_BMI",
      height < 130 ~ "possible_dwarfism",
      weight < 40 ~ "suspicious_low_weight",
      BMI < 15 | BMI > 60 ~ "suspicious_BMI",
      TRUE ~ "ok" # To catch all values that does not fit the criteria above
    )
  )

heart_data_cleaned <- heart_data_cleaned %>%
  filter(!(flag == "impossible_height" | flag == "impossible_weight" | flag == "impossible_BMI" )) # fi

summary(heart_data_cleaned)
```

```
##      index             id              age          gender        height
##  Min.   :    0   Min.   :    0   Min.   :10798   1:44729   Min.   :100.0
##  1st Qu.:17500   1st Qu.:25007   1st Qu.:17657   2:23940   1st Qu.:159.0
##  Median :35014   Median :50021   Median :19701             Median :165.0
##  Mean   :35002   Mean   :49976   Mean   :19464             Mean   :164.4
##  3rd Qu.:52487   3rd Qu.:74869   3rd Qu.:21324             3rd Qu.:170.0
##  Max.   :69999   Max.   :99999   Max.   :23713             Max.   :250.0
##      weight           ap_hi           ap_lo        cholesterol gluc
##  Min.   : 28.00   Min.   : 16.0   Min.   :  1.00   1:51496     1:58385
```

```
##  1st Qu.: 65.00   1st Qu.:120.0   1st Qu.: 80.00   2: 9301     2: 5068  
##  Median : 72.00   Median :120.0   Median : 80.00   3: 7872     3: 5216  
##  Mean   : 74.11   Mean   :126.7   Mean   : 81.27                        
##  3rd Qu.: 82.00   3rd Qu.:140.0   3rd Qu.: 90.00                        
##  Max.   :200.00   Max.   :240.0   Max.   :182.00                        
##  smoke     alco      active    cardio         BMI            flag          
##  0:62632   0:65007   0:13500   0:34698   Min.   :10.73   Length:68669      
##  1: 6037   1: 3662   1:55169   1:33971   1st Qu.:23.88   Class :character  
##                                          Median :26.35   Mode  :character  
##                                          Mean   :27.46                     
##                                          3rd Qu.:30.12                     
##                                          Max.   :74.38                     
```

Age

Now, if we look at the age column. It's stored in terms of days instead of years. It would make more sense and easier to analyze in terms of years. Let's convert age column to years.

```
heart_data_cleaned <- heart_data_cleaned %>%
  mutate(age = round((age / 365), digits = 0))

summary(heart_data_cleaned)
```

```
##      index             id             age          gender        height     
##  Min.   :    0   Min.   :    0   Min.   :30.00   1:44729   Min.   :100.0  
##  1st Qu.:17500   1st Qu.:25007   1st Qu.:48.00   2:23940   1st Qu.:159.0  
##  Median :35014   Median :50021   Median :54.00             Median :165.0  
##  Mean   :35002   Mean   :49976   Mean   :53.33             Mean   :164.4  
##  3rd Qu.:52487   3rd Qu.:74869   3rd Qu.:58.00             3rd Qu.:170.0  
##  Max.   :69999   Max.   :99999   Max.   :65.00             Max.   :250.0  
##      weight           ap_hi           ap_lo         cholesterol gluc     
##  Min.   : 28.00   Min.   : 16.0   Min.   :  1.00   1:51496     1:58385  
##  1st Qu.: 65.00   1st Qu.:120.0   1st Qu.: 80.00   2: 9301     2: 5068  
##  Median : 72.00   Median :120.0   Median : 80.00   3: 7872     3: 5216  
##  Mean   : 74.11   Mean   :126.7   Mean   : 81.27                        
##  3rd Qu.: 82.00   3rd Qu.:140.0   3rd Qu.: 90.00                        
##  Max.   :200.00   Max.   :240.0   Max.   :182.00                        
##  smoke     alco      active    cardio         BMI            flag          
##  0:62632   0:65007   0:13500   0:34698   Min.   :10.73   Length:68669      
##  1: 6037   1: 3662   1:55169   1:33971   1st Qu.:23.88   Class :character  
##                                          Median :26.35   Mode  :character  
##                                          Mean   :27.46                     
##                                          3rd Qu.:30.12                     
##                                          Max.   :74.38                     
```

Statistical Analysis

Correlation Testing Before we dive into analyzing process let's take a statistical approach and find correlation of which factors may increase the risk of cardiovascular heart disease.

```
# 1. Binary Target
cardio_binary <- as.numeric(heart_data_cleaned$cardio)

# 2. Numeric variables - correlation
numeric_vars <- c("age", "height", "weight", "ap_hi", "ap_lo", "BMI")
```

```
# 3. Calculate correlations and p-values
correlation_results <- sapply(numeric_vars, function(x) {
  # 3.1 Calculate correlation coefficient
  cor_coef <- cor(heart_data_cleaned[[x]], cardio_binary)

  # 3.2 Calculate p-value for correlation test
  cor_test <- cor.test(heart_data_cleaned[[x]], cardio_binary)

  return(c(correlation = cor_coef, p_value = cor_test$p.value))
})

# 4. Extract correlation coefficients and p-values
correlations <- correlation_results[1, ]
correlation_pvalues <- correlation_results[2, ]

# 5. Create correlation summary table
correlation_summary <- data.frame(
  Variable = numeric_vars,
  Correlation = round(correlations, 3),
  P_Value = round(correlation_pvalues, 4),
  Abs_Correlation = round(abs(correlations), 3),
  Significant = ifelse(correlation_pvalues < 0.05, "Yes", "No"),
  Direction = ifelse(correlations > 0, "Positive",
                   ifelse(correlations < 0, "Negative", "None")),
  Effect_Size = ifelse(abs(correlations) < 0.1, "Negligible",
                    ifelse(abs(correlations) < 0.3, "Small-Medium",
                          ifelse(abs(correlations) < 0.5, "Medium-Large", "Large")))
)

# 6. Categorical variables - chi-square and Cramér's V
categorical_vars <- c("gender", "cholesterol", "gluc", "smoke", "alco", "active")

# 7.Calculate both p-values and Cramér's V
categorical_results <- sapply(categorical_vars, function(x) {
  tbl <- table(heart_data_cleaned[[x]], heart_data_cleaned$cardio)
  chi_test <- chisq.test(tbl)

  # 7.1 Cramér's V calculation
  chi_sq <- chi_test$statistic
  n <- sum(tbl)
  min_dim <- min(dim(tbl)) - 1
  cramers_v <- sqrt(chi_sq / (n * min_dim))

  return(c(p_value = chi_test$p.value, cramers_v = cramers_v))
})

# 8. Extract results
chi_results <- categorical_results[1, ]
cramers_v <- categorical_results[2, ]

# 9. Create categorical summary
categorical_summary <- data.frame(
  Variable = categorical_vars,
```

```r
  P_Value = chi_results,
  Cramers_V = cramers_v,
  Significant = ifelse(chi_results < 0.05, "Yes", "No"),
  Effect_Size = ifelse(cramers_v < 0.1, "Small",
              ifelse(cramers_v < 0.3, "Medium", "Large"))
)
correlation_summary
```

```
##        Variable Correlation P_Value Abs_Correlation Significant Direction
## age         age       0.239  0.0000           0.239         Yes  Positive
## height   height      -0.011  0.0026           0.011         Yes  Negative
## weight   weight       0.180  0.0000           0.180         Yes  Positive
## ap_hi     ap_hi       0.428  0.0000           0.428         Yes  Positive
## ap_lo     ap_lo       0.336  0.0000           0.336         Yes  Positive
## BMI         BMI       0.189  0.0000           0.189         Yes  Positive
##         Effect_Size
## age    Small-Medium
## height   Negligible
## weight Small-Medium
## ap_hi  Medium-Large
## ap_lo  Medium-Large
## BMI    Small-Medium
```

```r
categorical_summary
```

```
##                Variable
## gender           gender
## cholesterol cholesterol
## gluc               gluc
## smoke             smoke
## alco               alco
## active           active
##
## gender      0.05827520196501902172148135150564485229551792144775390625000000000000000000000000000000
## cholesterol 0.00000000000000000000000000000000000000000000000000000000000000000000000000000000000000
## gluc        0.00000000000000000000000000000000000000000000000000000000000000000000000000000000000000
## smoke       0.00002044999005214192692786098759505364341748645529150962829589843750000000000000000000
## alco        0.02696930064458218784251286592734686564654111862182617187500000000000000000000000000000
## active      0.00000000000000000000988177029419894667407597254966825496334742514430642844226596651490
##               Cramers_V Significant Effect_Size
## gender      0.007226249          No       Small
## cholesterol 0.221502768         Yes      Medium
## gluc        0.091814155         Yes       Small
## smoke       0.016256277         Yes       Small
## alco        0.008441066         Yes       Small
## active      0.037448017         Yes       Small
```

Logistic Regression Analysis While correlation and Chi-square identify correlation at individual level, they do not explain how multiple risks level interact simultaneously. Logistic regression allows us to build a predictive model of cardiovascular heart disease presence (Yes/No), while estimating the relative importance of each variable.

1. Construct a model

```r
model <- glm(cardio ~ age + gender + height + weight + ap_hi + ap_lo + cholesterol + gluc + smoke + alco
             data = heart_data_cleaned,
             family = binomial)

summary(model)
```

```
##
## Call:
## glm(formula = cardio ~ age + gender + height + weight + ap_hi +
##     ap_lo + cholesterol + gluc + smoke + alco + active + BMI,
##     family = binomial, data = heart_data_cleaned)
##
## Coefficients:
##                  Estimate  Std. Error z value             Pr(>|z|)
## (Intercept)  -10.2699380   0.8656779 -11.863 < 0.0000000000000002 ***
## age            0.0507388   0.0013539  37.477 < 0.0000000000000002 ***
## gender2       -0.0146571   0.0221175  -0.663               0.5075
## height        -0.0069552   0.0052504  -1.325               0.1853
## weight         0.0140525   0.0055095   2.551               0.0108 *
## ap_hi          0.0563677   0.0009202  61.254 < 0.0000000000000002 ***
## ap_lo          0.0103811   0.0014330   7.244    0.000000000000434 ***
## cholesterol2   0.3787461   0.0273597  13.843 < 0.0000000000000002 ***
## cholesterol3   1.0973221   0.0357602  30.686 < 0.0000000000000002 ***
## gluc2          0.0194189   0.0362331   0.536               0.5920
## gluc3         -0.3500154   0.0395847  -8.842 < 0.0000000000000002 ***
## smoke1        -0.1443120   0.0348560  -4.140    0.000034695072728 ***
## alco1         -0.2180748   0.0424570  -5.136    0.000000280093665 ***
## active1       -0.2294351   0.0219063 -10.473 < 0.0000000000000002 ***
## BMI           -0.0082095   0.0145641  -0.564               0.5730
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 95188  on 68668  degrees of freedom
## Residual deviance: 76959  on 68654  degrees of freedom
## AIC: 76989
##
## Number of Fisher Scoring iterations: 4
```

2. Interpret results

```r
# Converting coefficients to odd-ratios

exp(coef(model))
```

```
##   (Intercept)           age       gender2        height        weight
## 0.00003465952 1.05204801199 0.98544974365 0.99306893040 1.01415170557
##         ap_hi         ap_lo  cholesterol2  cholesterol3         gluc2
## 1.05798661010 1.01043516176 1.46045218001 2.99613196462 1.01960869062
##         gluc3        smoke1         alco1       active1           BMI
## 0.70467727122 0.86561765185 0.80406528562 0.79498252265 0.99182407272
```

Based on the above result:

- age: Each additional year increases the odd by 5%

- gender2: Very close to one, this means the difference in terms of risk between male and female is minimal
- height: Each additional cm increase in height lowers the odd by 0.4%
- weight: Each additional kg increase in weight increases the odd by 1.1%
- ap_hi: Each additional increase in Systolic blood pressure reading increases the odd by 5.8%
- ap_lo: Each additional increase in Diatolic blood pressure reading increases the odd by 1%
- cholesterol2: When comparing to normal cholesterol level (1), moderately high cholesterol level increases the odd by 46%
- cholesterol3: When comparing to normal cholesterol level (1), high cholesterol level increases the odd by three-folds!
- gluc2: When comparing to normal glucose consumption level (1), medium glucose consumption increases the odd by 2%
- gluc3: When comparing to normal glucose consumption level (1), high glucose consumption lowers the odd by 30%.
- smoke1: When comparing to non-smokers (0), smoking lowers the odd by 14.5%
- alco1: When comparing to non-drinkers (0), drinking alcohol lowers the odd by 20.4 %
- active1: When comparing to low-activity (0), being active lowers the odd by 20.5%

Summary: Because this dataset is a synthetic dataset, some associations do not align with established medical evidence (eg. smoking lowers the odd of cardiovascular heart disease). Such results should be considered with caution.

From this Logistic Regression Model, cholesterol level, blood pressure reading, and weight are a strong predictor of cardiovascular heart disease. Cholesterol level has the highest effect where people with moderately high cholesterol level increases the odd of having cardiovascular disease by 46% and people with high cholesterol level increases the odd by three times. And by staying active, it lowers the odd by 20.5%!

3. Evaluating predictive accuracy of the model.

```r
# Retrieve predicted probabilities
prob <- predict(model, type = "response")

# Convert Probabilities to Classes
pred_class <- ifelse(prob > 0.5, 1, 0)

# Load caret library for machine learning tools
library(caret)

# Create confusion matrix
confusionMatrix(as.factor(pred_class), as.factor(heart_data_cleaned$cardio))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 27375 11357
##          1  7323 22614
##
##                Accuracy : 0.728
##                  95% CI : (0.7246, 0.7313)
##     No Information Rate : 0.5053
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 0.4552
##
##  Mcnemar's Test P-Value : < 0.00000000000000022
```

```
##
##             Sensitivity : 0.7890
##             Specificity : 0.6657
##          Pos Pred Value : 0.7068
##          Neg Pred Value : 0.7554
##              Prevalence : 0.5053
##          Detection Rate : 0.3987
##    Detection Prevalence : 0.5640
##       Balanced Accuracy : 0.7273
##
##        'Positive' Class : 0
##
```

4. Plot ROC curve & AUC

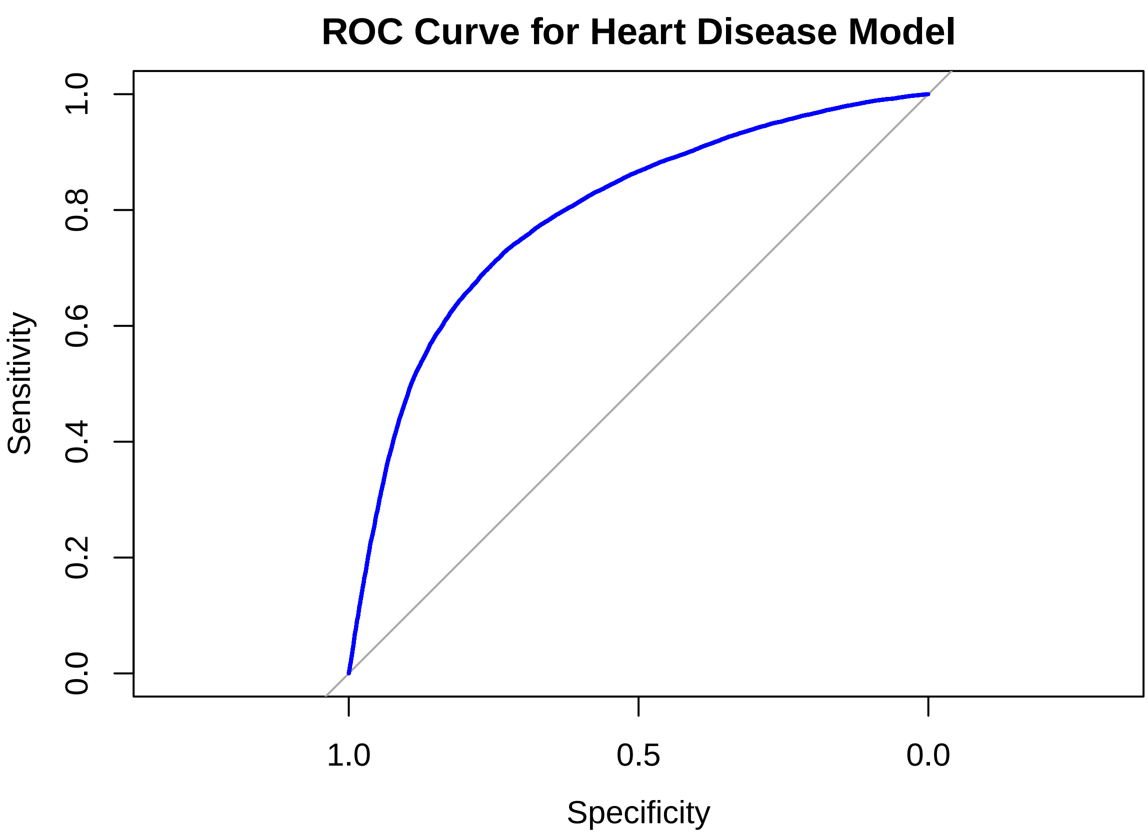
```r
library(pROC)

# Plot ROC Curve
roc_obj <- roc(heart_data_cleaned$cardio, prob)
```

```
## Setting levels: control = 0, case = 1
```

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

```r
plot(roc_obj, col = "blue", main = "ROC Curve for Heart Disease Model")
```




```r
# AUC (Area under curve)
auc_value <- auc(roc_obj)
auc_value
```

```
## Area under the curve: 0.7917
```

The model achieved an AUC of 0.79, suggesting a fair predictive ability.

5. Reduced Model Comparison

We remove variables that are counter-intuitive to medical evidence to compare whether this reduced model is a better predictive model.

```r
#Construct a reduced model (removing alcohol, gluc, and smoke)
reduced_model <- glm(cardio ~ age + gender + height + weight + ap_hi + ap_lo + cholesterol + active + BM
                     data = heart_data_cleaned,
                     family = binomial)

# Converting coefficient to odd-ratios
exp(coef(reduced_model))
```

```
##  (Intercept)          age      gender2       height       weight        ap_hi
## 0.0000355459 1.0521655074 0.9455985278 0.9929488899 1.0138044157 1.0578184803
##        ap_lo cholesterol2 cholesterol3      active1          BMI
## 1.0105493731 1.4444747167 2.5512129615 0.7923063137 0.9922220062
```

```r
# Retrieve predicted probabilities
prob_reduced <- predict(reduced_model, type = "response")

# Convert probabilities to class
pred_class_reduced <- ifelse(prob > 0.5, 1, 0)

# Construct confusion matrix
confusionMatrix(as.factor(pred_class_reduced), as.factor(heart_data_cleaned$cardio))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 27375 11357
##          1  7323 22614
##
##                Accuracy : 0.728
##                  95% CI : (0.7246, 0.7313)
##     No Information Rate : 0.5053
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 0.4552
##
##  Mcnemar's Test P-Value : < 0.00000000000000022
##
##             Sensitivity : 0.7890
##             Specificity : 0.6657
##          Pos Pred Value : 0.7068
##          Neg Pred Value : 0.7554
##              Prevalence : 0.5053
##          Detection Rate : 0.3987
##    Detection Prevalence : 0.5640
##       Balanced Accuracy : 0.7273
##
##        'Positive' Class : 0
```

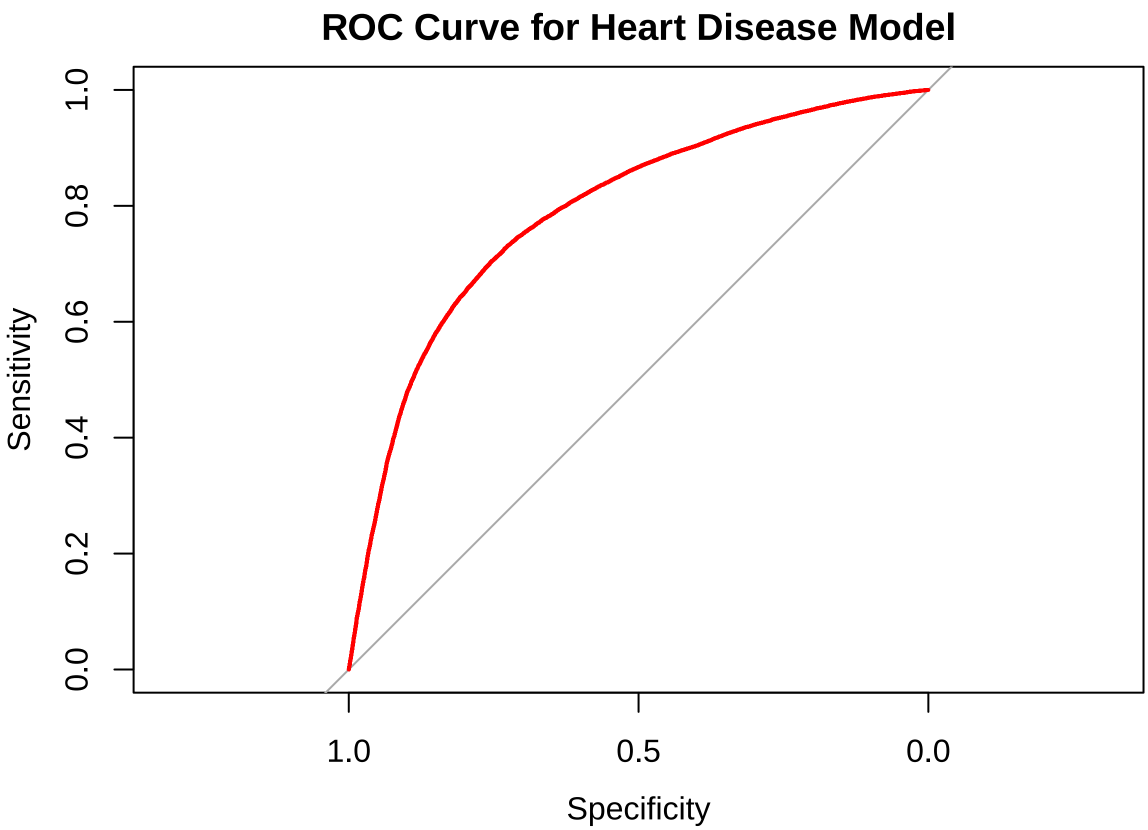
```
##
```

```r
# Plot ROC Curve
roc_obj_reduced <- roc(heart_data_cleaned$cardio, prob_reduced)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

```r
plot(roc_obj_reduced, col = "red", main = "ROC Curve for Heart Disease Model")
```




```r
# AUC (Area under curve)
auc_value_reduced <- auc(roc_obj_reduced)
auc_value_reduced
```

```
## Area under the curve: 0.7908
```

The initial logistic regression model achieved an accuracy of 72.8% with sensitivity of 78.9% and specificity of 66.6%. However, some predictors (e.g., smoking, alcohol, glucose) produced counterintuitive associations, likely due to the synthetic nature of the dataset.

To improve interpretability and potentially performance, we tested a reduced model excluding these variables. The reduced model achieved similar (or better) predictive performance, while providing more coherent interpretations of the remaining predictors.

As a result, the reduced model performs relatively the same with accuracy of 72.8%, sensitivity of 78.9%, and specificity of 66.6%. However, the AUC of the reduced model is neglectibly lower than the initial model (0.7908 and 0.7917 respectively). We conclude that initial model has a slightly better predictive capabilities than the reduced model.

Refinement & Interpretation **1. Multicollinearity Test**

```r
# Retrieve GVIF and Adjusted GVIF value to test for Multicollinearity
vif(model)
```

```
##                  GVIF Df GVIF^(1/(2*Df))
## age          1.023494  1        1.011679
## gender       1.459282  1        1.208008
## height      22.508905  1        4.744355
## weight      73.840118  1        8.593027
## ap_hi        1.770793  1        1.330712
## ap_lo        1.750250  1        1.322970
## cholesterol  1.498228  2        1.106355
## gluc         1.483421  2        1.103611
## smoke        1.246391  1        1.116419
## alco         1.140912  1        1.068135
## active       1.002701  1        1.001349
## BMI         69.993201  1        8.366194
```

The adjusted Generalized VIF is between 1-2 is considered no concern. When this value exceeding 5, it suggests that variables are redundant as they are highly related to each other and the model cannot diffferentiate between those variables, causing an unstable estimates.

Based on the result above, height, weight, and BMI are highly related to each other. Thus, the model is unable to differentiate their effects. We will exclude weight and height from the model and keep only BMI.

```r
# Exclude height and weight from the model
model <- glm(cardio ~ age + gender + ap_hi + ap_lo + cholesterol + gluc + smoke + alco + active + BMI,
             data = heart_data_cleaned,
             family = binomial)
# Test for GVIF
vif(model)
```

```
##                 GVIF Df GVIF^(1/(2*Df))
## age         1.016661  1        1.008296
## gender      1.150870  1        1.072786
## ap_hi       1.770403  1        1.330565
## ap_lo       1.748490  1        1.322305
## cholesterol 1.496350  2        1.106008
## gluc        1.482478  2        1.103436
## smoke       1.245859  1        1.116181
## alco        1.140818  1        1.068091
## active      1.002375  1        1.001187
## BMI         1.065943  1        1.032445
```

2. Re-testing the model's predictive capabilities

```r
# Converting coefficient to odd-ratios
exp(coef(model))
```

```
##   (Intercept)           age       gender2         ap_hi         ap_lo
## 0.00001169434 1.05151907239 1.03131466212 1.05808393740 1.01066571002
##  cholesterol2  cholesterol3         gluc2         gluc3        smoke1
## 1.46002648496 2.99439383167 1.02258426062 0.70688902792 0.86878461613
##         alco1       active1           BMI
## 0.80641952551 0.79394928595 1.02733727323
```

```r
# Retrieve predicted probabilities
prob <- predict(model, type = "response")
```

```r
# Convert probabilities to class
pred_class <- ifelse(prob > 0.5, 1, 0)

# Construct confusion matrix
confusionMatrix(as.factor(pred_class), as.factor(heart_data_cleaned$cardio))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 27393 11353
##          1  7305 22618
##
##                Accuracy : 0.7283
##                  95% CI : (0.7249, 0.7316)
##     No Information Rate : 0.5053
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 0.4558
##
##  Mcnemar's Test P-Value : < 0.00000000000000022
##
##             Sensitivity : 0.7895
##             Specificity : 0.6658
##          Pos Pred Value : 0.7070
##          Neg Pred Value : 0.7559
##              Prevalence : 0.5053
##          Detection Rate : 0.3989
##    Detection Prevalence : 0.5642
##       Balanced Accuracy : 0.7276
##
##        'Positive' Class : 0
##
```

```r
# AUC (Area under curve)
auc_value <- auc(roc_obj)
auc_value
```

```
## Area under the curve: 0.7917
```

Model performance remained stable (Accuracy = 72.8%, AUC = 0.79), suggesting that BMI adequately captures the relevant information without loss of predictive power. This confirms that simplifying the model improves interpretability while maintaining predictive ability.

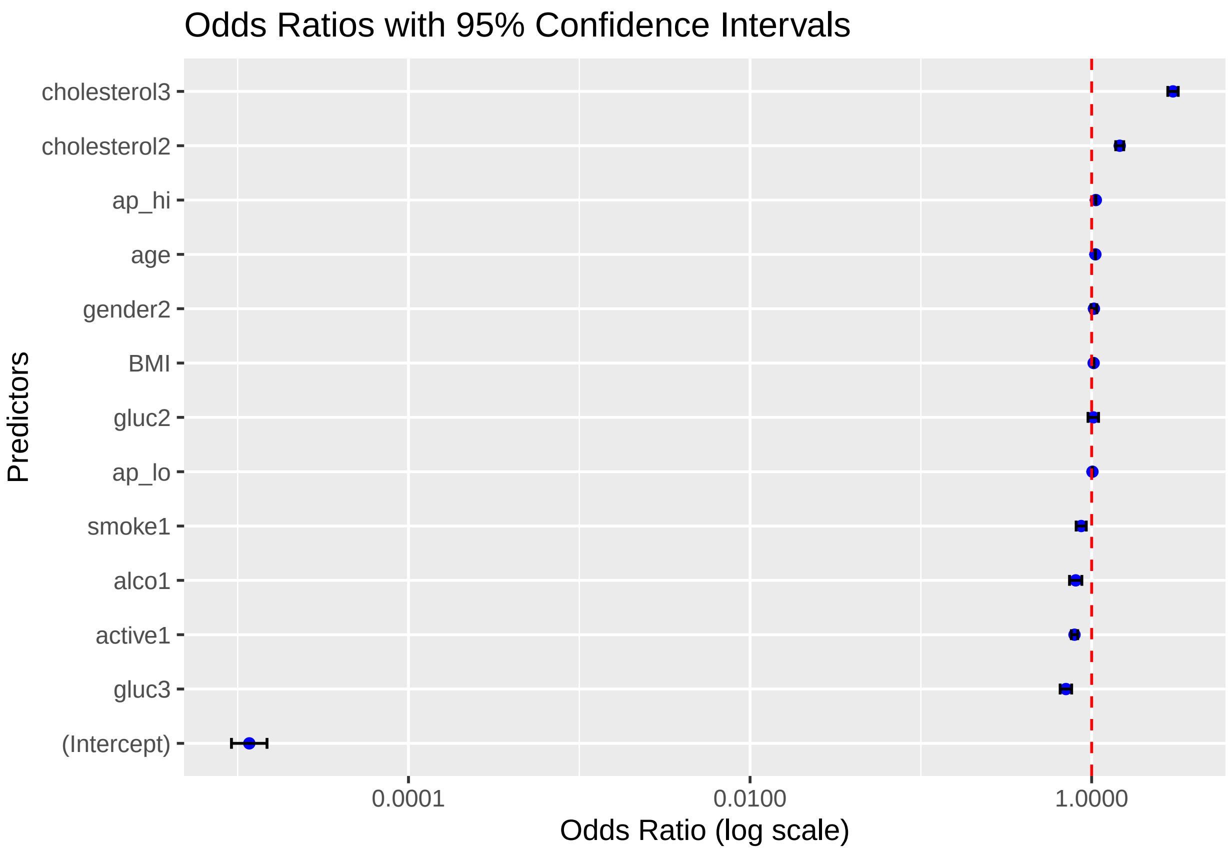
```r
library(broom)

# Convert model to tidy format
tidy_model <- tidy(model, exponentiate = TRUE, conf.int = TRUE)
```

Visualization 1. Odds ratios

```r
ggplot(tidy_model, aes(x = reorder(term, estimate), y = estimate)) +
  geom_point(color = "blue") +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high), width = 0.2) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "red") +
```

```r
  coord_flip() +
  labs(title = "Odds Ratios with 95% Confidence Intervals",
       x = "Predictors",
       y = "Odds Ratio (log scale)") +
  scale_y_log10()
```

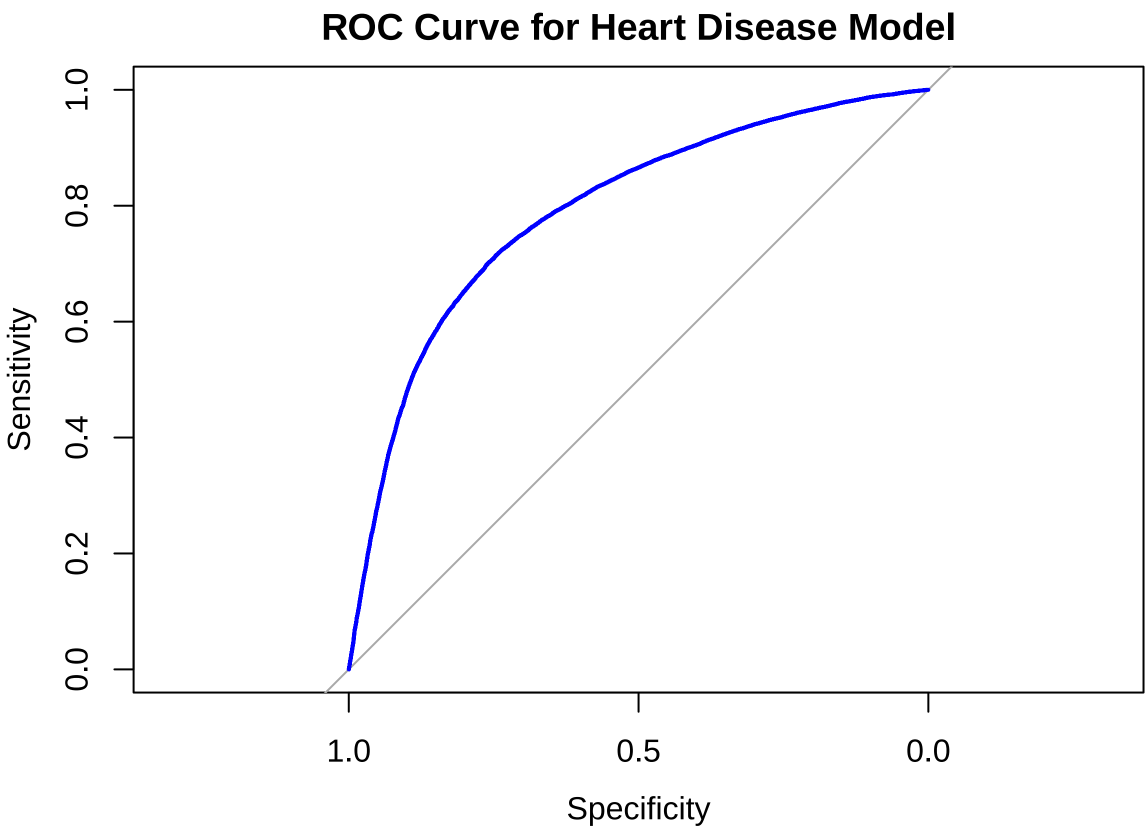


2. ROC Curve

```r
prob <- predict(model, type = "response")
roc_obj <- roc(heart_data_cleaned$cardio, prob)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
plot(roc_obj, col = "blue", main = "ROC Curve for Heart Disease Model")
```




```r
auc(roc_obj)
```

```
## Area under the curve: 0.7916
```

Limitation & Conclusion This project demonstrated how logistic regression can be applied to predict the likelihood of heart disease using a mix of demographic, lifestyle, and clinical variables. The model achieved an accuracy of approximately 73% and an AUC of 0.79, suggesting moderate predictive power.

Key predictors included age, blood pressure, and cholesterol levels, which consistently increased the odds of cardiovascular heart disease. In contrast, some lifestyle-related variables (smoking, alcohol, glucose) produced counterintuitive associations, reflecting the synthetic nature of the dataset. Another key indicator to look at is active, by staying active decreased the odds of cardiovascular heart disease by 20.6%.

Simplifying the model by addressing multicollinearity (retaining BMI while excluding height and weight) improved interpretability without reducing performance. Overall, logistic regression proved effective in identifying important predictors and providing interpretable results, though its predictive accuracy is limited compared to more advanced machine learning methods.

This work highlights the value of logistic regression as both a predictive and interpretive tool. While results from synthetic data should not be generalized to real-world patients, the modeling process demonstrates key steps in applied data analysis: variable exploration, model fitting, evaluation, and refinement.

References

- https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-readings
- https://www.ncbi.nlm.nih.gov/books/NBK535456/

The following packages were used in this analysis:

```r
library(pacman)

package_names <- c("rio", "here", "janitor", "lubridate", "matchmaker", "epikit", "tidyverse", "skimr",

lapply(package_names, p_citation)
```

```
## [[1]]
## To cite package 'rio' in publications use:
##
##   Chan C, Leeper T, Becker J, Schoch D (2023). _rio: A Swiss-army knife
##   for data file I/O_. <https://cran.r-project.org/package=rio>.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {rio: A Swiss-army knife for data file I/O},
##     author = {Chung-hong Chan and Thomas J. Leeper and Jason Becker and David Schoch},
##     url = {https://cran.r-project.org/package=rio},
##     year = {2023},
##   }
##
## [[2]]
## To cite package 'here' in publications use:
##
##   Müller K (2020). _here: A Simpler Way to Find Your Files_.
##   doi:10.32614/CRAN.package.here
##   <https://doi.org/10.32614/CRAN.package.here>, R package version
##   1.0.1, <https://CRAN.R-project.org/package=here>.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {here: A Simpler Way to Find Your Files},
##     author = {Kirill Müller},
##     year = {2020},
##     note = {R package version 1.0.1},
##     url = {https://CRAN.R-project.org/package=here},
##     doi = {10.32614/CRAN.package.here},
##   }
##
## [[3]]
## To cite package 'janitor' in publications use:
##
##   Firke S (2024). _janitor: Simple Tools for Examining and Cleaning
##   Dirty Data_. doi:10.32614/CRAN.package.janitor
##   <https://doi.org/10.32614/CRAN.package.janitor>, R package version
##   2.2.1, <https://CRAN.R-project.org/package=janitor>.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {janitor: Simple Tools for Examining and Cleaning Dirty Data},
##     author = {Sam Firke},
##     year = {2024},
```

```
##     note = {R package version 2.2.1},
##     url = {https://CRAN.R-project.org/package=janitor},
##     doi = {10.32614/CRAN.package.janitor},
##   }
##
## [[4]]
## To cite lubridate in publications use:
##
##   Garrett Grolemund, Hadley Wickham (2011). Dates and Times Made Easy
##   with lubridate. Journal of Statistical Software, 40(3), 1-25. URL
##   https://www.jstatsoft.org/v40/i03/.
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {Dates and Times Made Easy with {lubridate}},
##     author = {Garrett Grolemund and Hadley Wickham},
##     journal = {Journal of Statistical Software},
##     year = {2011},
##     volume = {40},
##     number = {3},
##     pages = {1--25},
##     url = {https://www.jstatsoft.org/v40/i03/},
##   }
##
## [[5]]
## To cite package 'matchmaker' in publications use:
##
##   Kamvar Z (2020). _matchmaker: Flexible Dictionary-Based Cleaning_.
##   doi:10.32614/CRAN.package.matchmaker
##   <https://doi.org/10.32614/CRAN.package.matchmaker>, R package version
##   0.1.1, <https://CRAN.R-project.org/package=matchmaker>.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {matchmaker: Flexible Dictionary-Based Cleaning},
##     author = {Zhian N. Kamvar},
##     year = {2020},
##     note = {R package version 0.1.1},
##     url = {https://CRAN.R-project.org/package=matchmaker},
##     doi = {10.32614/CRAN.package.matchmaker},
##   }
##
## [[6]]
## To cite package 'epikit' in publications use:
##
##   Spina A, Kamvar Z, Schumacher D (2024). _epikit: Miscellaneous Helper
##   Tools for Epidemiologists_. doi:10.32614/CRAN.package.epikit
##   <https://doi.org/10.32614/CRAN.package.epikit>, R package version
##   0.1.6, <https://CRAN.R-project.org/package=epikit>.
##
## A BibTeX entry for LaTeX users is
##
```

```
##   @Manual{,
##     title = {epikit: Miscellaneous Helper Tools for Epidemiologists},
##     author = {Alexander Spina and Zhian N. Kamvar and Dirk Schumacher},
##     year = {2024},
##     note = {R package version 0.1.6},
##     url = {https://CRAN.R-project.org/package=epikit},
##     doi = {10.32614/CRAN.package.epikit},
##   }
##
## [[7]]
## To cite package 'tidyverse' in publications use:
##
##   Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R,
##   Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller
##   E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V,
##   Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to
##   the tidyverse." _Journal of Open Source Software_, *4*(43), 1686.
##   doi:10.21105/joss.01686 <https://doi.org/10.21105/joss.01686>.
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {Welcome to the {tidyverse}},
##     author = {Hadley Wickham and Mara Averick and Jennifer Bryan and Winston Chang and Lucy D'Agosti
##     year = {2019},
##     journal = {Journal of Open Source Software},
##     volume = {4},
##     number = {43},
##     pages = {1686},
##     doi = {10.21105/joss.01686},
##   }
##
## [[8]]
## To cite package 'skimr' in publications use:
##
##   Waring E, Quinn M, McNamara A, Arino de la Rubia E, Zhu H, Ellis S
##   (2025). _skimr: Compact and Flexible Summaries of Data_.
##   doi:10.32614/CRAN.package.skimr
##   <https://doi.org/10.32614/CRAN.package.skimr>, R package version
##   2.2.1, <https://CRAN.R-project.org/package=skimr>.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {skimr: Compact and Flexible Summaries of Data},
##     author = {Elin Waring and Michael Quinn and Amelia McNamara and Eduardo {Arino de la Rubia} and
##     year = {2025},
##     note = {R package version 2.2.1},
##     url = {https://CRAN.R-project.org/package=skimr},
##     doi = {10.32614/CRAN.package.skimr},
##   }
##
## [[9]]
## To cite corrplot in publications use:
```

```
##
##   Taiyun Wei and Viliam Simko (2024). R package 'corrplot':
##   Visualization of a Correlation Matrix (Version 0.95). Available from
##   https://github.com/taiyun/corrplot
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{corrplot2024,
##     title = {R package 'corrplot': Visualization of a Correlation Matrix},
##     author = {Taiyun Wei and Viliam Simko},
##     year = {2024},
##     note = {(Version 0.95)},
##     url = {https://github.com/taiyun/corrplot},
##   }
##
## [[10]]
## To cite caret in publications use:
##
##   Kuhn, M. (2008). Building Predictive Models in R Using the caret
##   Package. Journal of Statistical Software, 28(5), 1-26.
##   https://doi.org/10.18637/jss.v028.i05
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {Building Predictive Models in R Using the caret Package},
##     volume = {28},
##     url = {https://www.jstatsoft.org/index.php/jss/article/view/v028i05},
##     doi = {10.18637/jss.v028.i05},
##     number = {5},
##     journal = {Journal of Statistical Software},
##     author = {{Kuhn} and {Max}},
##     year = {2008},
##     pages = {1-26},
##   }
##
## [[11]]
## If you use pROC in published research, please cite the following paper:
##
##   Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti,
##   Frédérique Lisacek, Jean-Charles Sanchez and Markus Müller (2011).
##   pROC: an open-source package for R and S+ to analyze and compare ROC
##   curves. BMC Bioinformatics, 12, p. 77.  DOI: 10.1186/1471-2105-12-77
##   <http://www.biomedcentral.com/1471-2105/12/77/>
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {pROC: an open-source package for R and S+ to analyze and compare ROC curves},
##     author = {Xavier Robin and Natacha Turck and Alexandre Hainard and Natalia Tiberti and Frédériqu
##     year = {2011},
##     journal = {BMC Bioinformatics},
##     volume = {12},
##     pages = {77},
```

```
##   }
##
## [[12]]
## To cite the car package in publications use:
##
##   Fox J, Weisberg S (2019). _An R Companion to Applied Regression_,
##   Third edition. Sage, Thousand Oaks CA.
##   <https://www.john-fox.ca/Companion/>.
##
## A BibTeX entry for LaTeX users is
##
##   @Book{,
##     title = {An {R} Companion to Applied Regression},
##     edition = {Third},
##     author = {John Fox and Sanford Weisberg},
##     year = {2019},
##     publisher = {Sage},
##     address = {Thousand Oaks {CA}},
##     url = {https://www.john-fox.ca/Companion/},
##   }
```