Data622_Assignment_4

Anna Moy & Natalie Kalukeerthie

2025-11-15

Heart Disease

Heart disease is one of the leading causes of death across the world due to bad eating habits. It is important healthcare experts are able to identify which patients have a higher risk in heart disease and provide early intervention and correct their lifestyle habits to avoid getting heart disease.

We used the dataset from Kaggle: Heart Disease Cleveland UCI. The dataset provides a list of patients information who went to Cleveland and determine if they had heart disease or not.

The problem we are trying to solve is: Predict whether a patient has heart disease based on clinical and demographic features. This is a binary classification as the target variable is heart disease(0) or no heart disease(1).

In this project, we evaluate four machine learning models—Logistic Regression, Random Forest, Support Vector Machine, and Neural Network—to determine which algorithm best predicts heart disease using clinical and demographic features. This comparison helps us understand which model performs best on small healthcare datasets.

```
#Load Libraries
library(tidyverse)
library(corrplot)
library(DataExplorer)
library(caret)
library(tidymodels)
library(themis)
library(caret)
library(randomForest)
library(yardstick)
library(dplyr)
library(e1071)
library(nnet)



# Import the CSV file from github
heart <- read_csv("https://raw.githubusercontent.com/AnnaMoy/Data-622/refs/heads/main/heart_cleveland_u
```

Exploratory Data Analysis

First step is to view the structure and look at the basic summary statistics of the dataset. In our dataset it shows we have 297 patient records and 14 features.

The EDA helped us identify potential outliers, relationships between the features, and which variables may influence heart disease risk. The correlation analysis highlighted thalach and oldpeak as strong predictors, which guided our focus when choosing algorithms and evaluating feature importance.

These are our features: Age(age), Sex(sex), Chest pain type(cp), Resting Blood Pressure(tresbps), Serum Cholestoral(chol), Fasting Blood sugar(fbs), Resting electrocardiographic results(restecg), Maximum heart rate achieved(thalach), Exercise induced angina(exang), ST depression induced by exercise relative to rest(oldpeak), the slope of the peak exercise ST segment(slope), number of major vessels(ca), (thal), heart disease or no heart disease (condition)

```
# View structure and basic summary
str(heart)
```

```
## spc_tbl_ [297 x 14] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ age      : num [1:297] 69 69 66 65 64 64 63 61 60 59 ...
##  $ sex      : num [1:297] 1 0 0 1 1 1 1 1 0 1 ...
##  $ cp       : num [1:297] 0 0 0 0 0 0 0 0 0 0 ...
##  $ trestbps : num [1:297] 160 140 150 138 110 170 145 134 150 178 ...
##  $ chol     : num [1:297] 234 239 226 282 211 227 233 234 240 270 ...
##  $ fbs      : num [1:297] 1 0 0 1 0 0 1 0 0 0 ...
##  $ restecg  : num [1:297] 2 0 0 2 2 2 2 0 0 2 ...
##  $ thalach  : num [1:297] 131 151 114 174 144 155 150 145 171 145 ...
##  $ exang    : num [1:297] 0 0 0 0 1 0 0 0 0 0 ...
##  $ oldpeak  : num [1:297] 0.1 1.8 2.6 1.4 1.8 0.6 2.3 2.6 0.9 4.2 ...
##  $ slope    : num [1:297] 1 0 2 1 1 1 2 1 0 2 ...
##  $ ca       : num [1:297] 1 2 0 1 0 0 0 2 0 0 ...
##  $ thal     : num [1:297] 0 0 0 0 0 2 1 0 0 2 ...
##  $ condition: num [1:297] 0 0 0 1 0 0 0 1 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   age = col_double(),
##   ..   sex = col_double(),
##   ..   cp = col_double(),
##   ..   trestbps = col_double(),
##   ..   chol = col_double(),
##   ..   fbs = col_double(),
##   ..   restecg = col_double(),
##   ..   thalach = col_double(),
##   ..   exang = col_double(),
##   ..   oldpeak = col_double(),
##   ..   slope = col_double(),
##   ..   ca = col_double(),
##   ..   thal = col_double(),
##   ..   condition = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
summary(heart)
```

```
##       age             sex               cp           trestbps
##  Min.   :29.00   Min.   :0.0000   Min.   :0.000   Min.   : 94.0
##  1st Qu.:48.00   1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:120.0
##  Median :56.00   Median :1.0000   Median :2.000   Median :130.0
```

```
##  Mean   :54.54   Mean   :0.6768   Mean   :2.158   Mean   :131.7
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:3.000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :3.000   Max.   :200.0
##       chol            fbs             restecg          thalach
##  Min.   :126.0   Min.   :0.0000   Min.   :0.0000   Min.   : 71.0
##  1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:133.0
##  Median :243.0   Median :0.0000   Median :1.0000   Median :153.0
##  Mean   :247.4   Mean   :0.1448   Mean   :0.9966   Mean   :149.6
##  3rd Qu.:276.0   3rd Qu.:0.0000   3rd Qu.:2.0000   3rd Qu.:166.0
##  Max.   :564.0   Max.   :1.0000   Max.   :2.0000   Max.   :202.0
##      exang           oldpeak          slope              ca
##  Min.   :0.0000   Min.   :0.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.800   Median :1.0000   Median :0.0000
##  Mean   :0.3266   Mean   :1.056   Mean   :0.6027   Mean   :0.6768
##  3rd Qu.:1.0000   3rd Qu.:1.600   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :6.200   Max.   :2.0000   Max.   :3.0000
##       thal         condition
##  Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000
##  Median :0.000   Median :0.0000
##  Mean   :0.835   Mean   :0.4613
##  3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :2.000   Max.   :1.0000
```

We checked to see if there was any missing data and there were no missing data. There were no duplicate in our dataset. We looked at the correlation between the five numerical features.

The correlation between age vs. thalach is -.39 which tells us as people age their maximum heart rate will decrease.

The correlation between thalach vs. oldpeak is -.35 which indicates as ones maximum heart rate increases the ST depression will be less.

```
# Check for NA
colSums(is.na(heart))
```

```
##       age       sex        cp  trestbps      chol       fbs   restecg   thalach
##         0         0         0         0         0         0         0         0
##     exang   oldpeak     slope        ca      thal condition
##         0         0         0         0         0         0
```

```
num_duplicates <- sum(duplicated(heart))
cat("Number of duplicate rows:", num_duplicates, "\n")
```

```
## Number of duplicate rows: 0
```

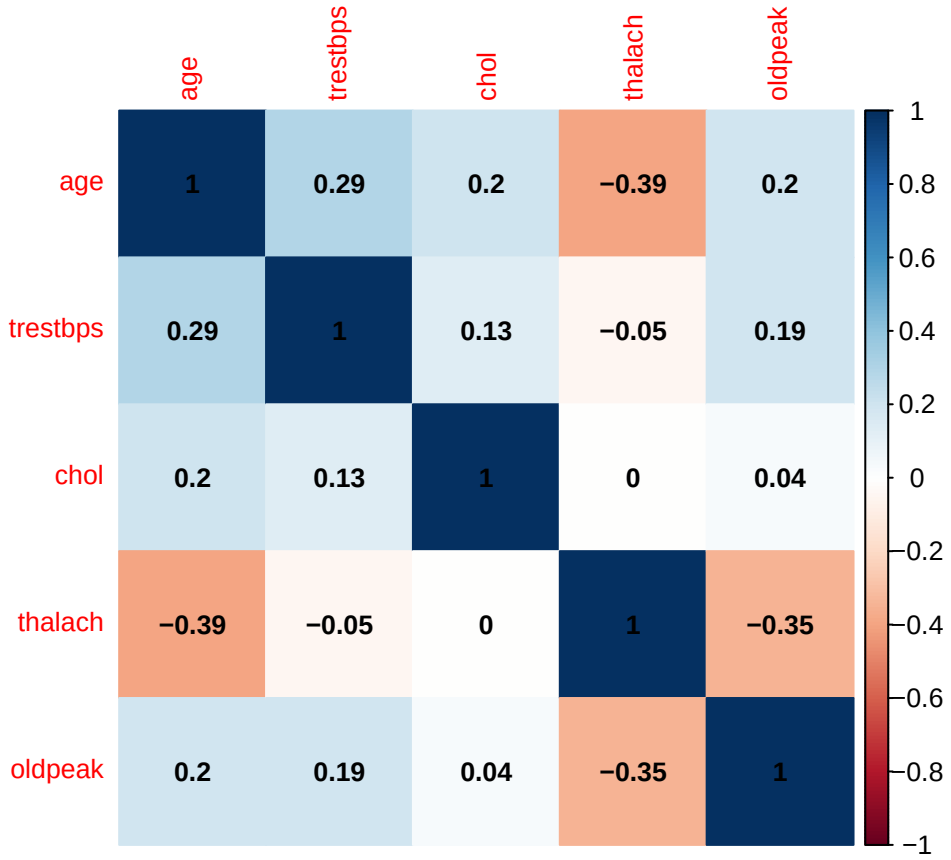
```
# Select numeric columns
num_cols <- heart %>% select(age, trestbps, chol, thalach, oldpeak)

# Compute correlation matrix
cor_matrix <- cor(num_cols)

# View correlation matrix
print(cor_matrix)
```

```
##                 age    trestbps          chol       thalach     oldpeak
## age       1.0000000  0.29047626  2.026435e-01 -3.945629e-01  0.19712262
## trestbps  0.2904763  1.00000000  1.315357e-01 -4.910766e-02  0.19124314
## chol      0.2026435  0.13153571  1.000000e+00 -7.456799e-05  0.03859579
## thalach  -0.3945629 -0.04910766 -7.456799e-05  1.000000e+00 -0.34763997
## oldpeak   0.1971226  0.19124314  3.859579e-02 -3.476400e-01  1.00000000
```

```r
# Plot full correlation heatmap with numbers in all boxes
corrplot(cor_matrix,
         method = "color",      # colored squares
         type = "full",         # show full matrix (both triangles)
         addCoef.col = "black",# add correlation coefficients
         tl.cex = 0.8,          # text label size
         number.cex = 0.8,      # coefficient text size
         diag = TRUE)           # include diagonal
```



The features like age, resting blood pressure and cholesterol have weaker correlations and less of a predictor. With maximum heart rate and ST depression are better predictors.

```r
cor_with_target <- sapply(num_cols, function(x) cor(x, heart$condition))
print(cor_with_target)
```

```
##         age    trestbps        chol     thalach     oldpeak
##  0.22707515  0.15349003  0.08028475 -0.42381706  0.42405206
```

Calculated the mean, median, standard deviation and interquartile range (IQR) for the numeric variable. With the median and mean fairly close to each other indicates there may be some outliers to the data.

```
num_cols %>% summarise_all(list(mean = mean, median = median, sd = sd, IQR = IQR))
```

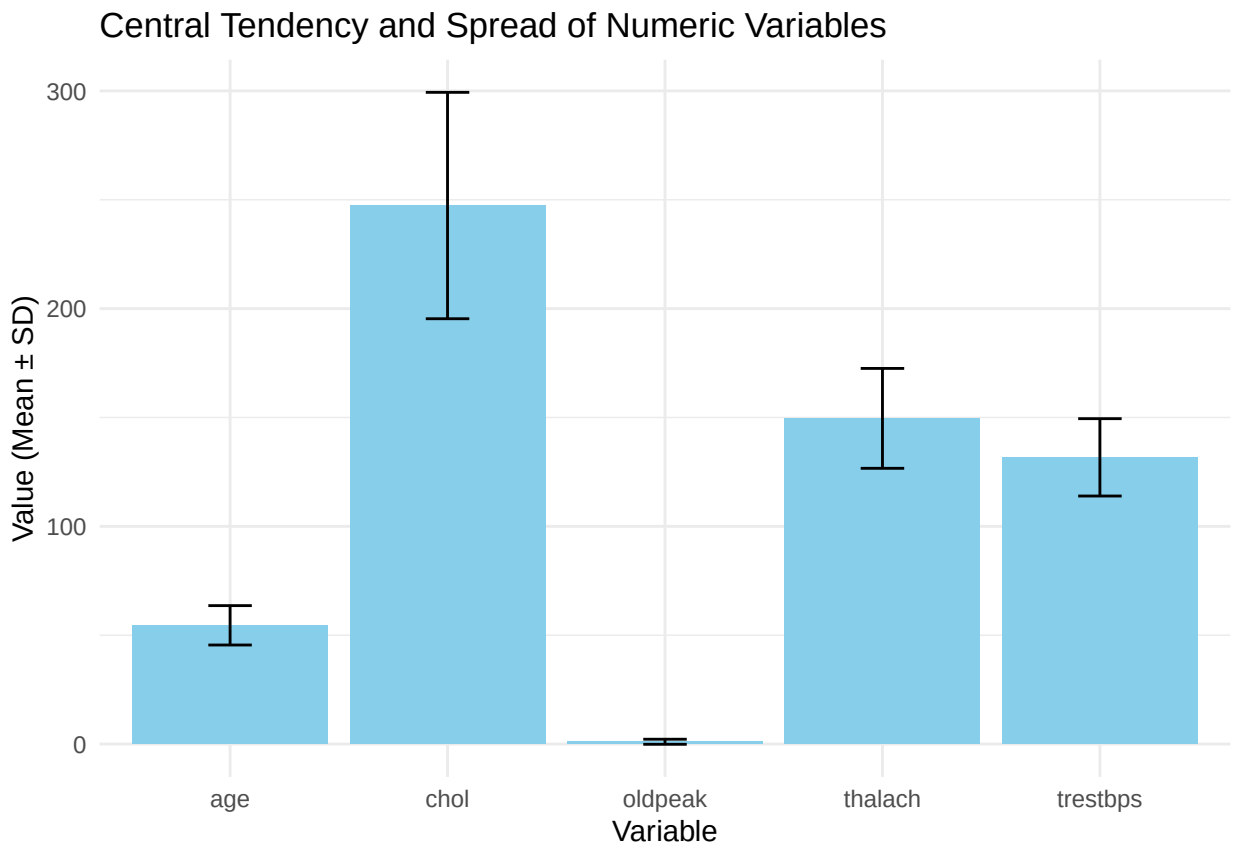
```
## # A tibble: 1 x 20
##   age_mean trestbps_mean chol_mean thalach_mean oldpeak_mean age_median
##      <dbl>         <dbl>     <dbl>        <dbl>        <dbl>      <dbl>
## 1     54.5          132.      247.         150.         1.06         56
## # i 14 more variables: trestbps_median <dbl>, chol_median <dbl>,
## #   thalach_median <dbl>, oldpeak_median <dbl>, age_sd <dbl>,
## #   trestbps_sd <dbl>, chol_sd <dbl>, thalach_sd <dbl>, oldpeak_sd <dbl>,
## #   age_IQR <dbl>, trestbps_IQR <dbl>, chol_IQR <dbl>, thalach_IQR <dbl>,
## #   oldpeak_IQR <dbl>
```

```
# Compute summary statistics
summary_stats <- num_cols %>%
  summarise_all(list(
    mean = ~mean(.),
    median = ~median(.),
    sd = ~sd(.),
    min = ~min(.),
    max = ~max(.)
  )) %>%
  # Reshape for plotting
  pivot_longer(everything(), names_to = c("variable","stat"), names_sep = "_") %>%
  pivot_wider(names_from = stat, values_from = value)

# View table
print(summary_stats)
```

```
## # A tibble: 5 x 6
##   variable   mean median    sd   min   max
##   <chr>     <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 age       54.5    56    9.05    29  77
## 2 trestbps 132.    130   17.8     94 200
## 3 chol     247.    243   52.0    126 564
## 4 thalach  150.    153   22.9     71 202
## 5 oldpeak    1.06    0.8  1.17     0   6.2
```

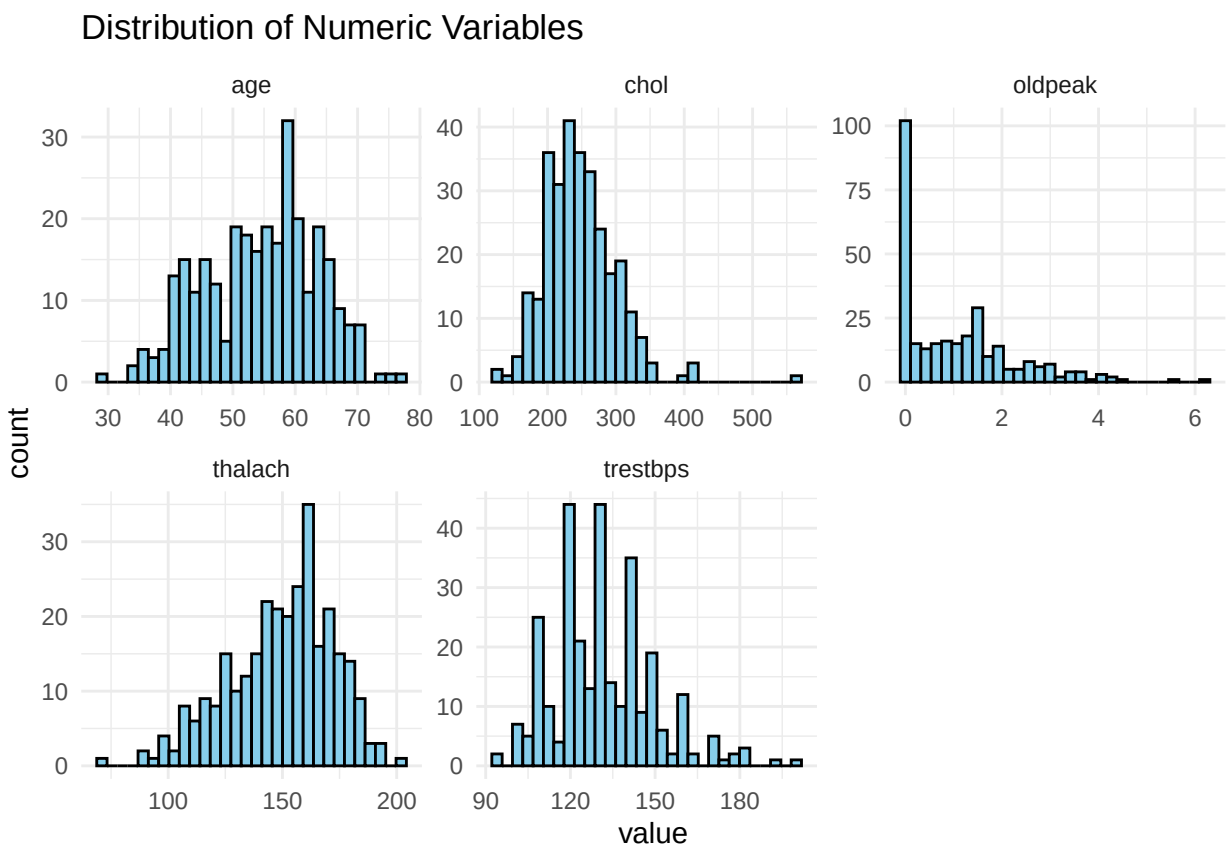
```
# Plot mean and sd for easy visualization
summary_stats %>%
  ggplot(aes(x = variable)) +
  geom_bar(aes(y = mean), stat = "identity", fill = "skyblue") +
  geom_errorbar(aes(ymin = mean - sd, ymax = mean + sd), width = 0.2) +
  labs(title = "Central Tendency and Spread of Numeric Variables",
       y = "Value (Mean ± SD)",
       x = "Variable") +
  theme_minimal()
```



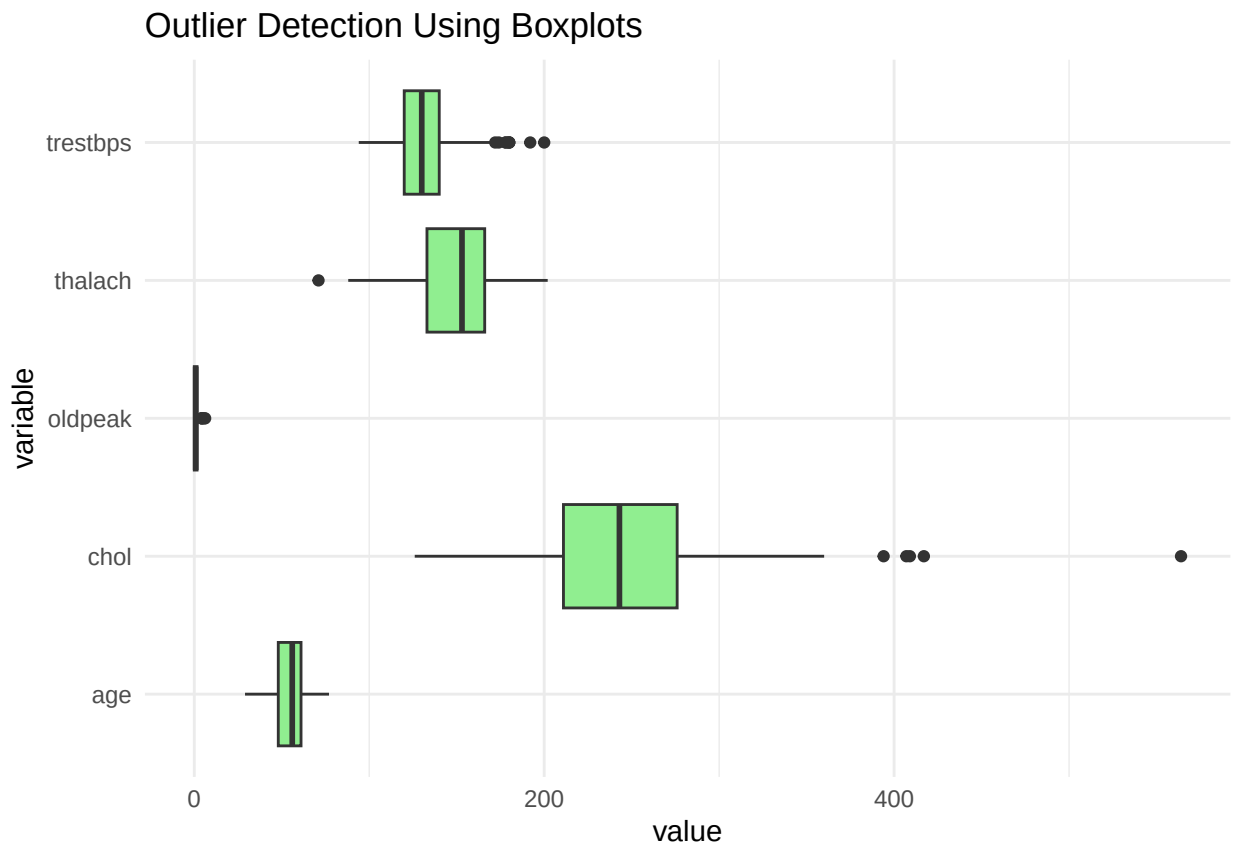
The distribution of the dataset shows age is normally distributed which means there is a diversity in the patient among their age. With cholestoral(chol) right skewed it shows many patients have cholestoral between 200-300 and it is rarely above 350+. The maximum heart rate is slightly left skewed with maximum heart rates around 125- 150 indicating a lot of patients are exercising or doing some type of activity.

With the boxplot we were able to see where the outliers were for the different features. The serum cholestoral and resting blood pressure has the most outliers. There are a few patients with their cholestoral at a high level.

```
# Plot histograms for all numeric variables
num_cols %>%
  gather(key = "variable", value = "value") %>%
  ggplot(aes(x = value)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  facet_wrap(~ variable, scales = "free") +
  theme_minimal() +
  labs(title = "Distribution of Numeric Variables")
```




```r
# Boxplot to check for outliers
num_cols %>%
  gather(key = "variable", value = "value") %>%
  ggplot(aes(x = variable, y = value)) +
  geom_boxplot(fill = "lightgreen") +
  theme_minimal() +
  labs(title = "Outlier Detection Using Boxplots") +
  coord_flip()
```



For the categorical data we looked at the counts and proportion for each feature. There were 67% of the patients were male and 47% were asymptomatic to chest pain.

```r
# Check levels and counts for categorical columns
categorical_cols <- c("sex", "cp", "fbs", "restecg", "exang", "slope", "ca", "thal")

# List of label mappings for categorical variables
labels_list <- list(
  sex = c("0" = "Female", "1" = "Male"),
  cp = c("0" = "Typical Angina", "1" = "Atypical Angina", "2" = "Non-anginal Pain", "3" = "Asymptomatic
  fbs = c("0" = "Fasting BS <= 120 mg/dl", "1" = "Fasting BS > 120 mg/dl"),
  restecg = c("0" = "Normal", "1" = "ST-T wave abnormality", "2" = "Left ventricular hypertrophy"),
  exang = c("0" = "No Exercise Induced Angina", "1" = "Exercise Induced Angina"),
  slope = c("0" = "Upsloping", "1" = "Flat", "2" = "Downsloping"),
  ca = c("0" = "0 Vessels", "1" = "1 Vessel", "2" = "2 Vessels", "3" = "3 Vessels"),
  thal = c("0" = "Normal", "1" = "Fixed Defect", "2" = "Reversable Defect"),
  condition = c("0" = "No Heart Disease", "1" = "Heart Disease")
)

# Function to print counts with labels for each variable
for (var in names(labels_list)) {
  cat("\nVariable:", var, "\n")
  counts <- table(heart[[var]])
  names(counts) <- labels_list[[var]][names(counts)]
  print(counts)
  props <- prop.table(counts)
```

```
  print(round(props, 3))
}
```

```
##
## Variable: sex
## Female   Male
##     96    201
## Female   Male
##  0.323  0.677
##
## Variable: cp
##   Typical Angina  Atypical Angina Non-anginal Pain     Asymptomatic
##               23               49               83              142
##   Typical Angina  Atypical Angina Non-anginal Pain     Asymptomatic
##            0.077            0.165            0.279            0.478
##
## Variable: fbs
## Fasting BS <= 120 mg/dl  Fasting BS > 120 mg/dl
##                     254                     43
## Fasting BS <= 120 mg/dl  Fasting BS > 120 mg/dl
##                   0.855                   0.145
##
## Variable: restecg
##                        Normal        ST-T wave abnormality
##                           147                            4
## Left ventricular hypertrophy
##                           146
##                        Normal        ST-T wave abnormality
##                         0.495                        0.013
## Left ventricular hypertrophy
##                         0.492
##
## Variable: exang
## No Exercise Induced Angina    Exercise Induced Angina
##                        200                         97
## No Exercise Induced Angina    Exercise Induced Angina
##                      0.673                      0.327
##
## Variable: slope
##   Upsloping        Flat Downsloping
##         139         137          21
##   Upsloping        Flat Downsloping
##       0.468       0.461       0.071
##
## Variable: ca
## 0 Vessels  1 Vessel 2 Vessels 3 Vessels
##       174        65        38        20
## 0 Vessels  1 Vessel 2 Vessels 3 Vessels
##     0.586     0.219     0.128     0.067
##
## Variable: thal
##            Normal      Fixed Defect Reversable Defect
##               164                18               115
```

```
##            Normal      Fixed Defect Reversable Defect
##             0.552             0.061             0.387
##
## Variable: condition
## No Heart Disease    Heart Disease
##              160              137
## No Heart Disease    Heart Disease
##            0.539            0.461
```

With the focus on gender we looked at the number of patients with and without heart disease. Female had a less chance of heart disease compared to male patients.

```
# Create labeled factors
heart$sex_label <- factor(heart$sex, levels = c(0,1), labels = c("Female", "Male"))
heart$condition_label <- factor(heart$condition, levels = c(0,1), labels = c("No Heart Disease", "Heart

# Contingency table with labels
sex_condition_table <- table(heart$sex_label, heart$condition_label)
print(sex_condition_table)
```

```
##
##          No Heart Disease Heart Disease
##   Female               71            25
##   Male                 89           112
```

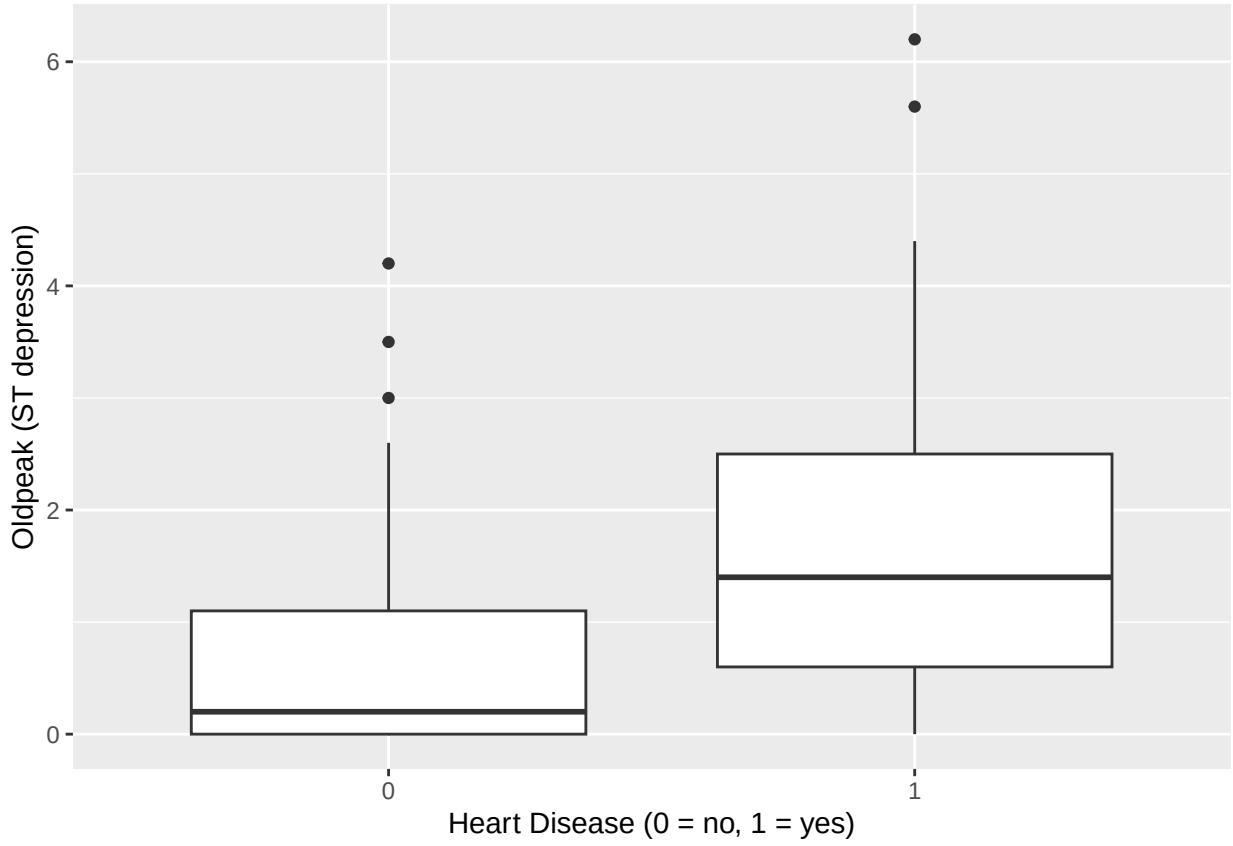
```
# Proportions by row (sex)
sex_condition_props <- prop.table(sex_condition_table, margin = 1)
print(round(sex_condition_props, 3))
```

```
##
##          No Heart Disease Heart Disease
##   Female            0.740         0.260
##   Male              0.443         0.557
```

We looked to see if there was any patterns or trends in our data. In the boxplot you can see that as a patient has ST depression they were more likely to have heart disease.

In the scatter chart showing maximum heart rate vs. age the heart rate is lower as they age. People with heart disease tends to have a higher maximum heart rate and primarily between 30-55 years of age.

```
# Boxplot of oldpeak by heart disease outcome
ggplot(heart, aes(x = factor(condition), y = oldpeak)) +
  geom_boxplot() +
  labs(x = "Heart Disease (0 = no, 1 = yes)", y = "Oldpeak (ST depression)")
```




```
# Scatter plot of age vs thalach colored by condition
ggplot(heart, aes(x = age, y = thalach, color = factor(condition))) +
  geom_point(alpha = 0.6) +
  labs(color = "Heart Disease")
```

Preprocessing

As identified earlier, cholesterol (chol) and resting blood pressure (trestbps) have extreme outliers. Since outliers can distort model performance, we will limit their influence by Winsorizing, keeping values within the 1st to 99th percentile. This way, extreme values are capped rather than removed, preserving important information for analysis.

We chose Winsorizing because removing outliers would shrink an already small dataset. By capping extreme values, we preserved all patient records while reducing distortion. Scaling was required because algorithms like SVM and Neural Networks are sensitive to the magnitude of numeric features.

```
# Handling outliers
# Winsorize cholesterol and trestbps to replace extreme outliers
heart$chol <- pmin(pmax(heart$chol, quantile(heart$chol, 0.01)), quantile(heart$chol, 0.99))
heart$trestbps <- pmin(pmax(heart$trestbps, quantile(heart$trestbps, 0.01)), quantile(heart$trestbps, 0
```

We are creating a new feature by grouping patients' ages into buckets to see if age categories reveal any patterns. Additionally, we are combining age and ST depression (oldpeak) into a new feature to explore whether age has an effect on ST depression levels, which may help improve the model's predictive power.

```
#feature engineering
heart <- heart %>%
  mutate(
    age_group = case_when(
```

```
      age <= 45 ~ "30-45",
      age <= 60 ~ "46-60",
      TRUE ~ "61+"
    ),
    age_oldpeak = age * oldpeak
  )
```

For the numeric features we selected age, resting blood pressure, cholestoral, maximum heart rate, depression and age range. For the categorical feature we picked only a few which are sex, chest pain and exercise induced agina and then changing them to factors. These will be the features we use to run our algorithm.

```
set.seed(123)

# Convert target to factor BEFORE split
heart$condition <- factor(heart$condition)

# Select features
num_cols <- c("age", "trestbps", "chol", "thalach", "oldpeak", "age_group","age_oldpeak")
cat_cols <- c("sex", "cp", "exang")  # pick 2-3 categorical features

# Ensure categorical are factors
heart[cat_cols] <- lapply(heart[cat_cols], factor)
```

Machine Learning Algorithm

We split the dataset into training and testing sets using an 80/20 ratio. Then, we identified the numeric features and applied scaling. The model was trained using a 5-fold cross -validation.

```
# Split dataset
trainIndex <- createDataPartition(heart$condition, p = 0.8, list = FALSE)
train <- heart[trainIndex, ]
test  <- heart[-trainIndex, ]

# Scale numeric features
preProc <- preProcess(train[, num_cols], method = c("center", "scale"))
train[, num_cols] <- predict(preProc, train[, num_cols])
test[, num_cols] <- predict(preProc, test[, num_cols])
```

Logistic Regression

Logistic Regression was selected as it is great for binary classification problem (heart disease or no heart disease). The algorithm is widely used, easy to interpret, simple and fast and ideal for small dataset. The downfall in selecting a logistic regression is it struggles on complex patterns and sensitive to variables.

With the dataset with only 297 patients it is ideal to use logistic regression and we can add in each feature to see how it affects the likelihood of the patient having heart disease.

The results of the logistic regression had an accuracy of 77.6% which is not bad but would be great if we can improve it. The recall indicates the correctly identify patients without heart disease 84% of the time.

Accuracy: 77% of the patients were classified correct Precision: 77% of the patients with heart disease actually had heart disease Recall: 84% of patients who actually have heart disease were correctly identified F1 Score: 80% reflects the balance of precision and recall

```
# Logistic Regression
log_model <- train(condition ~ age + trestbps + chol + thalach + oldpeak + age_group+ age_oldpeak + sex
                   data = train,
                   method = "glm",
                   family = "binomial",
                   trControl = trainControl(method = "cv", number = 5))  # 5-fold CV

# Predictions
pred_log <- predict(log_model, test)

# Confusion matrix
conf_mat <- confusionMatrix(pred_log, test$condition)
print(conf_mat)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 27  8
##          1  5 19
##
##                Accuracy : 0.7797
##                  95% CI : (0.6527, 0.8771)
##     No Information Rate : 0.5424
##     P-Value [Acc > NIR] : 0.0001366
##
##                   Kappa : 0.5522
##
##  Mcnemar's Test P-Value : 0.5790997
##
##             Sensitivity : 0.8438
##             Specificity : 0.7037
##          Pos Pred Value : 0.7714
##          Neg Pred Value : 0.7917
##              Prevalence : 0.5424
##          Detection Rate : 0.4576
##    Detection Prevalence : 0.5932
##       Balanced Accuracy : 0.7737
##
##        'Positive' Class : 0
##
```

```
# Get accuracy, precision, recall, F1 in a table
metrics <- data.frame(
  Accuracy  = conf_mat$overall["Accuracy"],
  Precision = conf_mat$byClass["Pos Pred Value"],
  Recall    = conf_mat$byClass["Sensitivity"],
  F1        = 2 * ((conf_mat$byClass["Pos Pred Value"] * conf_mat$byClass["Sensitivity"]) /
                   (conf_mat$byClass["Pos Pred Value"] + conf_mat$byClass["Sensitivity"]))
```

```
)
print(metrics)
```

```
##          Accuracy Precision  Recall        F1
## Accuracy 0.779661 0.7714286 0.84375 0.8059701
```

Logistic Regression Model With More Features

Since the Logistic Regression model had fairly good results in accuracy we decided to add additional features to see if this would results in better accuracy and results. The additional features we added were age and ST depression(age_oldpeak), sex(sex), chest pain(cp), exercise induced angina(exang), peak exercise ST segment(slope),number of major vessels(ca), thal(0-normal,1-fixed defect, 2-reversable defect),resting electrocardiographic(restecg),fasting blood sugar(fbs).

The results of this model was better than the previous logistic regression model due to the additional features added. Overall the model performance increased by approximately 5% overall in the different metrics.

Accuracy: 83% of the patients were classified correct Precision: 82% of the patients with heart disease actually had heart disease Recall: 87% of patients who actually have heart disease were correctly identified F1 Score: 84% reflects the balance of precision and recall

```
# Improved Logistic Regression Model (with more features)
log_model2 <- train(
  condition ~ age + trestbps + chol + thalach + oldpeak +
    age_group + age_oldpeak + sex + cp + exang +
    slope + ca + thal + restecg + fbs,
  data = train,
  method = "glm",
  family = "binomial",
  trControl = trainControl(method = "cv", number = 5)
)

# Predictions on the test set
pred_log2 <- predict(log_model2, test)

# Confusion Matrix
conf_mat <- confusionMatrix(pred_log2, test$condition)
print(conf_mat)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 28  6
##          1  4 21
##
##                Accuracy : 0.8305
##                  95% CI : (0.7103, 0.9156)
##     No Information Rate : 0.5424
##     P-Value [Acc > NIR] : 3.14e-06
##
##                   Kappa : 0.6566
```

```
##
##  Mcnemar's Test P-Value : 0.7518
##
##             Sensitivity : 0.8750
##             Specificity : 0.7778
##          Pos Pred Value : 0.8235
##          Neg Pred Value : 0.8400
##              Prevalence : 0.5424
##          Detection Rate : 0.4746
##    Detection Prevalence : 0.5763
##       Balanced Accuracy : 0.8264
##
##        'Positive' Class : 0
##
```

```r
# Print metrics: Accuracy, Precision, Recall, F1
metrics2 <- data.frame(
  Accuracy  = conf_mat$overall["Accuracy"],
  Precision = conf_mat$byClass["Pos Pred Value"],
  Recall    = conf_mat$byClass["Sensitivity"],
  F1        = 2 * (
    (conf_mat$byClass["Pos Pred Value"] * conf_mat$byClass["Sensitivity"]) /
    (conf_mat$byClass["Pos Pred Value"] + conf_mat$byClass["Sensitivity"])
  )
)

print(metrics2)
```

```
##           Accuracy Precision Recall        F1
## Accuracy 0.8305085 0.8235294  0.875 0.8484848
```

Random Forest Model

Random Forest is using many decision trees on random subset of features and it is great for non-linear relationship, great for numeric and categorial data, and less sensitive to outliers. The downside to using random forest is that it can easily overfit and be less intepretable.

We wanted to see if there was an increase in performance if we used the Random Forest model using the same features for Logistic Regression.

The results of the model did not outperform the Logistic Regression model. With the small dataset it was not model did not perform as well. The accuracy was 69% and the precision, recall and F1 score was the same at 72% in this model.

Accuracy: 69% of the patients were classified correct Precision: 72% of the patients with heart disease actually had heart disease Recall: 72% of patients who actually have heart disease were correctly identified F1 Score: 72% reflects the balance of precision and recall

```r
# Random Forest
rf_model <- train(condition ~ age + trestbps + chol + thalach + oldpeak + age_group+ age_oldpeak +
                    sex + cp + exang,
                  data = train,
                  method = "rf",
```

```
                  trControl = trainControl(method = "cv", number = 5),
                  tuneLength = 5)  # smaller grid helps reduce overfitting

# Predictions
pred_rf <- predict(rf_model, test)

# Evaluate
conf_mat_rf <- confusionMatrix(pred_rf, test$condition)
print(conf_mat_rf)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 24 10
##          1  8 17
##
##                Accuracy : 0.6949
##                  95% CI : (0.5613, 0.8081)
##     No Information Rate : 0.5424
##     P-Value [Acc > NIR] : 0.01224
##
##                   Kappa : 0.3818
##
##  Mcnemar's Test P-Value : 0.81366
##
##             Sensitivity : 0.7500
##             Specificity : 0.6296
##          Pos Pred Value : 0.7059
##          Neg Pred Value : 0.6800
##              Prevalence : 0.5424
##          Detection Rate : 0.4068
##    Detection Prevalence : 0.5763
##       Balanced Accuracy : 0.6898
##
##        'Positive' Class : 0
##
```

```
# Get accuracy, precision, recall, F1 in a table
metrics_rf <- data.frame(
  Accuracy = conf_mat_rf$overall["Accuracy"],
  Precision = conf_mat_rf$byClass["Pos Pred Value"],
  Recall = conf_mat_rf$byClass["Sensitivity"],
  F1 = 2 * ((conf_mat_rf$byClass["Pos Pred Value"] * conf_mat_rf$byClass["Sensitivity"]) /
              (conf_mat_rf$byClass["Pos Pred Value"] + conf_mat_rf$byClass["Sensitivity"]))
)
print(metrics)
```

```
##          Accuracy Precision  Recall        F1
## Accuracy 0.779661 0.7714286 0.84375 0.8059701
```

Support Vector Machine (SVM)

Support Vector Machine is best for separate classes. It is great for those with high dimensional space and works well with small dataset. It can be sensitive to scaling of the features and less interpretable.

We used a third model for comparison as this model works better with small dataset.The results for this was better than the Random Forest but not better than the Logistic Regression Model.

Accuracy: 76% of the patients were classified correct Precision: 75% of the patients with heart disease actually had heart disease Recall: 85% of patients who actually have heart disease were correctly identified F1 Score: 80% reflects the balance of precision and recall

SVM performed better than Random Forest because it works well with small datasets and can separate classes effectively in high-dimensional feature space. However, it still could not surpass Logistic Regression since the dataset did not benefit significantly from non-linear boundaries.

```
### ---- SELECT FEATURES ----
heart2 <- heart %>%
  select(condition, age, trestbps, chol, thalach, oldpeak,age_group, age_oldpeak,
         cp, thal, exang)

# Convert categorical to factors
heart2$cp <- factor(heart2$cp)
heart2$thal <- factor(heart2$thal)
heart2$exang <- factor(heart2$exang)
heart2$condition <- factor(heart2$condition, levels = c(0,1))

### ---- TRAIN/TEST SPLIT ----
set.seed(123)
trainIndex <- createDataPartition(heart2$condition, p = 0.8, list = FALSE)
train <- heart2[trainIndex, ]
test  <- heart2[-trainIndex, ]

### ---- DUMMY ENCODE CATEGORICALS (NO LEAKAGE) ----
dummies <- dummyVars(condition ~ ., data = train)

train_x <- predict(dummies, newdata = train)
test_x  <- predict(dummies, newdata = test)

# Add target back
train_y <- train$condition
test_y  <- test$condition

### ---- SCALE NUMERICAL FEATURES ----
num_cols <- c("age", "trestbps", "chol", "thalach", "oldpeak", "age_oldpeak")

preProc <- preProcess(train_x[, num_cols], method = c("center", "scale"))

train_x[, num_cols] <- predict(preProc, train_x[, num_cols])
test_x[, num_cols]  <- predict(preProc, test_x[, num_cols])

### ---- TRAIN SVM MODEL ----
set.seed(123)
svm_model <- svm(train_x, train_y, kernel = "radial", cost = 1, gamma = 0.1)
```

```r
### ---- PREDICT ----
pred_svm <- predict(svm_model, test_x)

### ---- CREATE METRIC TABLE ----
conf <- confusionMatrix(pred_svm, test_y)

results <- data.frame(
  Accuracy  = conf$overall["Accuracy"],
  Precision = conf$byClass["Pos Pred Value"],
  Recall    = conf$byClass["Sensitivity"],
  F1        = conf$byClass["F1"]
)

print(results)
```

```
##           Accuracy Precision  Recall        F1
## Accuracy 0.7627119      0.75 0.84375 0.7941176
```

Neural Network Classification

Neural Network was used as a fourth model to determine if the results would be better than the Logistic Regression model. This is great for complex and nonlinear relationship but it is not great for small datasets.

Accuracy: 62% of the patients were classified correct Precision: 63% of the patients with heart disease actually had heart disease Recall: 72% of patients who actually have heart disease were correctly identified F1 Score: 67% reflects the balance of precision and recall

Neural Networks typically require thousands of records to learn complex patterns. With only 297 samples, the model struggled to generalize. This underperformance highlights that deep learning methods are not always ideal for small, structured datasets.

```r
set.seed(123)
# Split dataset
trainIndex <- createDataPartition(heart$condition, p = 0.8, list = FALSE)
train_nn <- heart[trainIndex, ]
test_nn  <- heart[-trainIndex, ]

# Scale numeric features (correct!)
preProc <- preProcess(train_nn[, num_cols], method = c("center", "scale"))
train_nn[, num_cols] <- predict(preProc, train_nn[, num_cols])
test_nn[, num_cols]  <- predict(preProc, test_nn[, num_cols])

# Train neural network (correct!)
nn_model <- train(
  condition ~ age + trestbps + chol + thalach + oldpeak + age_oldpeak +
    sex + cp + exang,
  data = train_nn,
  method = "nnet",
  trControl = trainControl(method = "cv", number = 5),
  tuneGrid = expand.grid(size = c(3, 5), decay = c(0.01, 0.1)),
  maxit = 500,
  trace = FALSE
```

```r
)

# Predictions (correct!)
pred_nn <- predict(nn_model, test_nn)

# Fix factor levels
pred_nn <- factor(pred_nn, levels = levels(test_nn$condition))

# Confusion matrix
conf_mat_nn <- confusionMatrix(pred_nn, test_nn$condition)
print(conf_mat_nn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 23 13
##          1  9 14
##
##                Accuracy : 0.6271
##                  95% CI : (0.4915, 0.7496)
##     No Information Rate : 0.5424
##     P-Value [Acc > NIR] : 0.1194
##
##                   Kappa : 0.24
##
##  Mcnemar's Test P-Value : 0.5224
##
##             Sensitivity : 0.7188
##             Specificity : 0.5185
##          Pos Pred Value : 0.6389
##          Neg Pred Value : 0.6087
##              Prevalence : 0.5424
##          Detection Rate : 0.3898
##    Detection Prevalence : 0.6102
##       Balanced Accuracy : 0.6186
##
##        'Positive' Class : 0
##
```

```r
# Metrics
metrics_nn <- data.frame(
  Accuracy  = conf_mat_nn$overall["Accuracy"],
  Precision = conf_mat_nn$byClass["Pos Pred Value"],
  Recall    = conf_mat_nn$byClass["Sensitivity"],
  F1        = 2 * ((conf_mat_nn$byClass["Pos Pred Value"] * conf_mat_nn$byClass["Sensitivity"]) /
             (conf_mat_nn$byClass["Pos Pred Value"] + conf_mat_nn$byClass["Sensitivity"]))
)
print(metrics_nn)
```

```
##           Accuracy Precision  Recall        F1
## Accuracy 0.6271186 0.6388889 0.71875 0.6764706
```

```r
# Combine all model metrics into one table
results_summary <- data.frame(
  Model = c(
    "Logistic Regression",
    "Logistic Regression (More Features)",
    "Random Forest",
    "SVM",
    "Neural Network"
  ),
  Accuracy  = c(metrics$Accuracy, metrics2$Accuracy, metrics_rf$Accuracy, results$Accuracy, metrics_nn$A
  Precision = c(metrics$Precision, metrics2$Precision, metrics_rf$Precision, results$Precision, metrics_
  Recall    = c(metrics$Recall, metrics2$Recall, metrics_rf$Recall, results$Recall, metrics_nn$Recall),
  F1        = c(metrics$F1, metrics2$F1, metrics_rf$F1, results$F1, metrics_nn$F1)
)

# Round only numeric columns
results_summary[,-1] <- round(results_summary[,-1], 3)

print(results_summary)
```

```
##                                Model Accuracy Precision Recall    F1
## 1                Logistic Regression    0.780     0.771  0.844 0.806
## 2 Logistic Regression (More Features)    0.831     0.824  0.875 0.848
## 3                      Random Forest    0.695     0.706  0.750 0.727
## 4                                SVM    0.763     0.750  0.844 0.794
## 5                     Neural Network    0.627     0.639  0.719 0.676
```

Conclusion

The Logistic Regression model with additional features performed the best overall with an accuracy of 83%, outperforming the Logistic Regression, Random Forest, SVM, and Neural Network models. This suggests that the relationship between the clinical features and heart disease risk is largely linear and that the dataset may be too small for more complex models to improve performance.

Across all models, features such as ST depression (oldpeak), maximum heart rate (thalach), chest pain type,age, age and ST depression(age_oldpeak), sex(sex), chest pain(cp), exercise induced angina(exang), peak exercise ST segment(slope),number of major vessels(ca), thal(0-normal,1-fixed defect, 2-reversable defect),resting electrocardiographic(restecg),fasting blood sugar(fbs) showed the strongest influence on predicting heart disease. These results reinforce established medical research—patients with abnormal ST depression levels, lower maximum heart rate, and certain chest pain categories are more likely to have heart disease.

While the models performed reasonably well, performance could likely be improved with:

- A larger dataset to reduce variance and improve model generalization
- Additional feature engineering (interaction terms, polynomial features)
- Hyperparameter tuning, especially for SVM and Neural Network models
- Oversampling techniques (SMOTE) if class imbalance is present
- More medical features that influence heart disease risk

Overall, this project demonstrates that machine learning can help identify high-risk patients using routine clinical data. Early intervention and lifestyle modification can be targeted toward individuals with higher predicted risk, improving healthcare outcomes and enabling preventative treatment strategies.