Statitical Models Homework 1

In this report, we will analyze a dataset using R and the Quarto framework. The analysis will involve data loading, cleaning and visualization, as well as statistical modeling and evaluation. To achieve this, we will use several essential R packages:

- `ggplot2` for data visualization
- `caret` for machine learning and model evaluation
- `tidyverse` for data manipulation and transformation
- `tidyr` for data tidying
- `gridExtra` for arranging multiple plots
- `patchwork` to group more plots in a grid
- `class` to work with knn model

Loading and Cleaning

The first step in our analysis is to load and inspect the dataset. We will check for missing values (NA) and determine the best approach to handle them. If the number of missing values is small, we may choose to remove them. Otherwise, we will consider imputing them using statistical techniques such as mean, median, or mode replacement. This step ensures that our dataset is clean and ready for further analysis.

```
[1] "Number of NA values: 204"
```

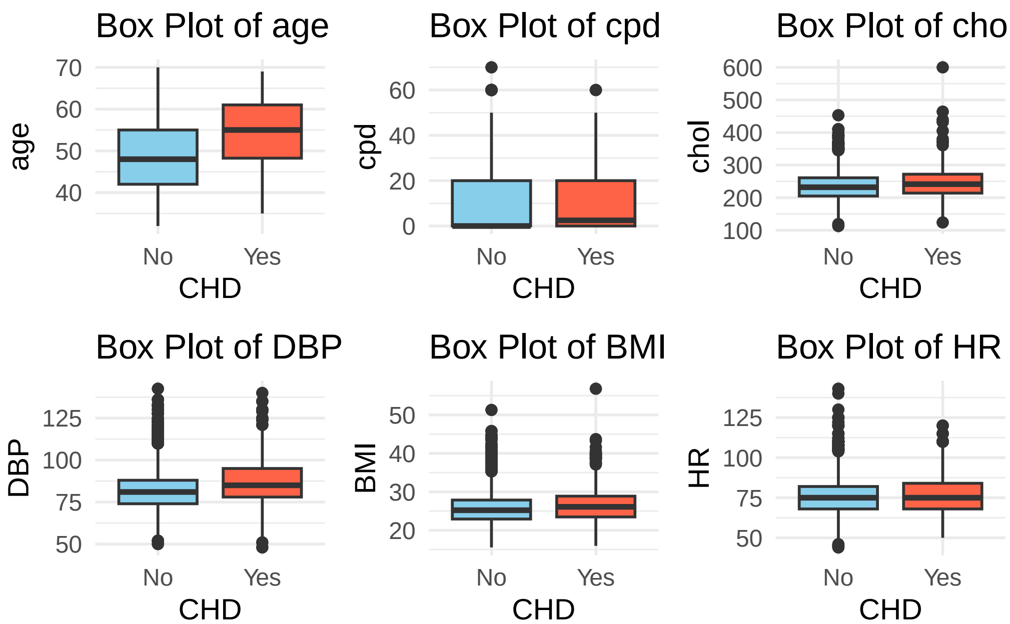
```
[1] "number of NA in education 105"
```

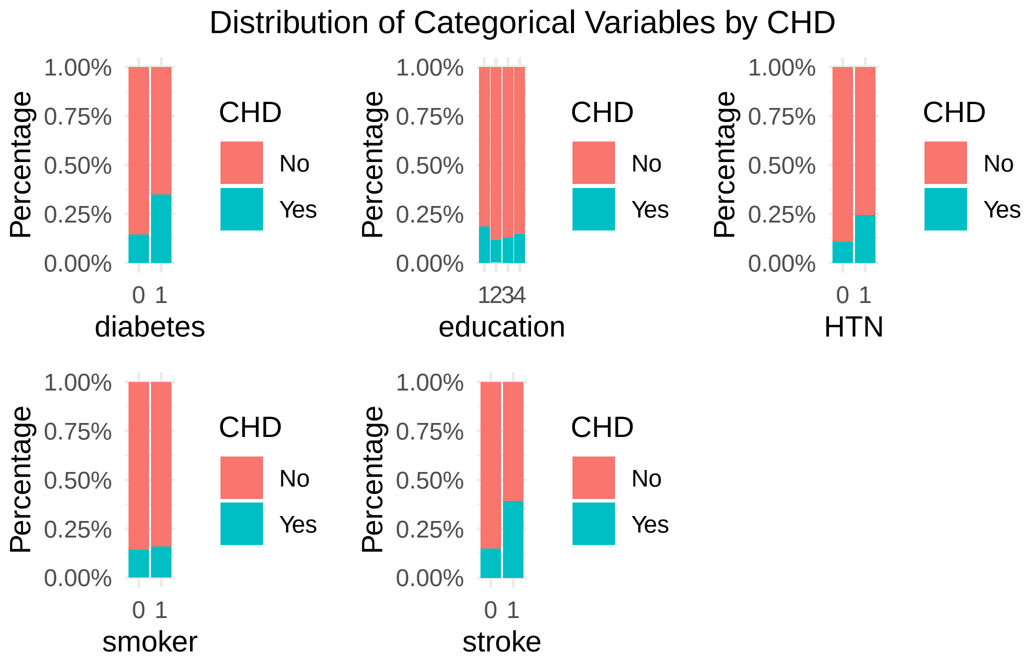
```
[1] "Proportion between yes and no"
```

```
       No       Yes
0.8499629 0.1500371
```

First of all we load the chd.csv, look at the NA values are 204, so i choose to delete them because of the small weight on the total dataset, it is also noticeable that more than half of the missing values are in the education column. Looking at the dataset, it is evident that the variable CHD is binary, with values 'No' and 'Yes', the "sex" is the other one with type character, the issues that can be encountered in the analysis are related to the inbalance of the dataset, as seen in the output the proportion of No in CHD column is 85%.

We proceed with visualizing plots that illustrate the conditional distribution based on the presence of our target variable. We will first examine the numerical variables, followed by the categorical ones.





Data Splitting and model decision

The code starts by setting a random seed to make sure the results can be reproduced by others. It then uses the createDataPartition function from the caret package to split the data into training (80%) and testing (20%) sets while maintaining the same proportion of CHD cases in both sets. This is important because heart disease is relatively rare in the dataset, so we need to make sure both sets have enough positive examples.

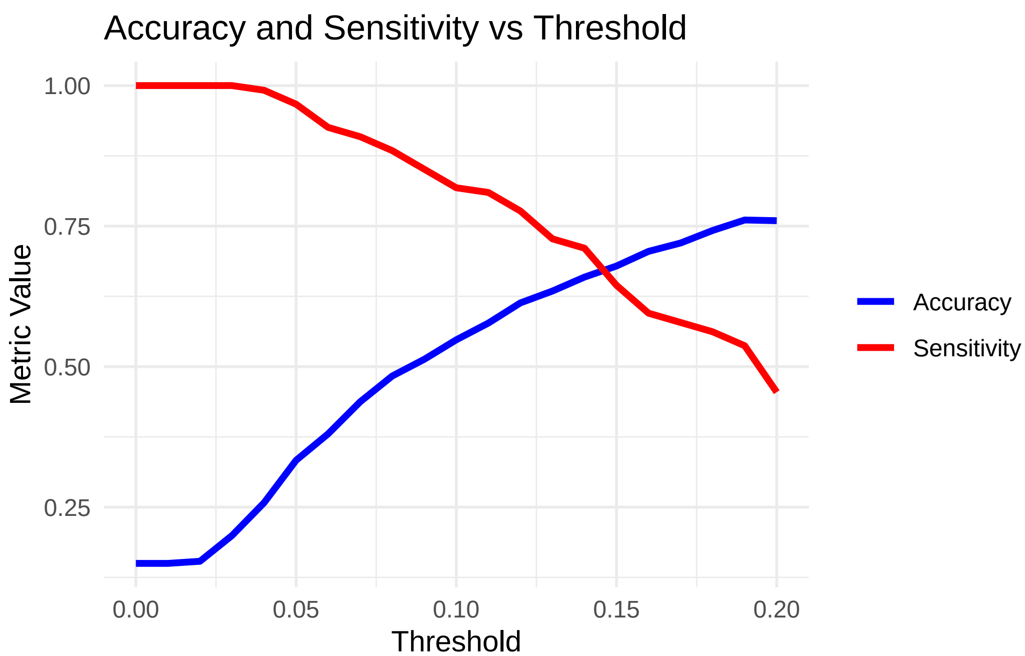
What's particularly smart here is the exploration of different thresholds between 0 and 0.2. In medical settings, we typically want to avoid missing cases of heart disease (false negatives), even if it means some healthy people might get additional tests (false positives). By testing these lower thresholds instead of the standard 0.5, the researchers are acknowledging that in this context, it's better to be cautious and catch more potential heart disease cases, even at the cost of some false alarms.

This threshold testing shows good understanding of both the statistical needs and the practical clinical considerations in heart disease prediction.

```
set.seed(123)


# Create a balanced split of the data (80% training, 20%
# testing)
index <- createDataPartition(df$CHD, p = 0.8, list = FALSE)
```

```r
train_data <- df[index, ]

test_data <- df[-index, ]
glm_model <- glm(CHD ~ ., data = train_data, family = binomial)
prob_pred <- predict(glm_model, test_data, type = "response")

# Step 2: Define thresholds
thresholds <- seq(0, 0.2, by = 0.01)
```

To visualize in a clearer way the choice of the threshold, we plotted the accuracy and sensitivity trends along different thresholds



We first verified the distribution of CHD cases in our training and test datasets to ensure proper representation. The model summary helped us identify key risk factors associated with heart disease. We carefully formatted our output variables as factors with consistent levels to ensure accurate evaluation metrics. Most notably, we applied a custom threshold of 0.14 instead of the standard 0.5 based on our previous sensitivity analysis. This lower threshold reflects our priority of identifying more potential CHD cases, accepting some false positives as a reasonable trade-off in a medical context. Finally, we created a confusion matrix for a comprehensive performance evaluation, giving us a complete picture of correct predictions and error types. This approach provides more meaningful insights than simple accuracy, especially for medical applications where missing a diagnosis can have serious consequences.

```
  No  Yes 
2747  485 


 No Yes 
686 121 


Call:
glm(formula = CHD ~ ., family = binomial, data = train_data)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept) -7.743e+00  7.346e-01 -10.540  < 2e-16 ***
sexMale      3.844e-01  1.132e-01   3.395 0.000685 ***
age          7.009e-02  6.887e-03  10.178  < 2e-16 ***
education   -3.489e-02  5.272e-02  -0.662 0.508151    
smoker       1.501e-01  1.647e-01   0.912 0.362006    
cpd          1.777e-02  6.551e-03   2.712 0.006693 ** 
stroke       8.486e-01  5.012e-01   1.693 0.090462 .  
HTN          4.339e-01  1.349e-01   3.217 0.001297 ** 
diabetes     5.960e-01  2.615e-01   2.279 0.022660 *  
chol         1.867e-03  1.174e-03   1.590 0.111854    
DBP          1.530e-02  5.374e-03   2.846 0.004424 ** 
BMI          4.268e-03  1.319e-02   0.324 0.746190    
HR           5.345e-05  4.459e-03   0.012 0.990437    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2733.1  on 3231  degrees of freedom
Residual deviance: 2463.1  on 3219  degrees of freedom
AIC: 2489.1

Number of Fisher Scoring iterations: 5

Confusion Matrix and Statistics

          Reference
Prediction  No Yes
```

```
       No  413  28
       Yes 273  93

               Accuracy : 0.627
                 95% CI : (0.5926, 0.6605)
    No Information Rate : 0.8501
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2021

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.6020
            Specificity : 0.7686
         Pos Pred Value : 0.9365
         Neg Pred Value : 0.2541
             Prevalence : 0.8501
         Detection Rate : 0.5118
   Detection Prevalence : 0.5465
      Balanced Accuracy : 0.6853

       'Positive' Class : No
```

The following code performs predictions using the k-NN model with cross validation with 5 folds and computes the confusion matrix we set a seed(123) for reproducibility and we look for the best k from 1 to 40, found being 35. the output shows an higher accuracy but fails to identify any CHD cases (Sensitivity = 0%). The Kappa of 0 and significant McNemar's test confirm the model is no better than random guessing for the minority class

```r
# Scaling data
preproc <- preProcess(train_features, method = c("center", "scale"))
train_features_scaled <- predict(preproc, train_features)
test_features_scaled <- predict(preproc, test_features)

## Tuning k with cross-validation
ctrl <- trainControl(method = "cv", number = 5, classProbs = TRUE,
    summaryFunction = twoClassSummary)

# Grid of k values to test
k_grid <- expand.grid(k = seq(1, 40, by = 2))
```

```r
# train of the model with tuning
knn_fit <- train(x = train_features_scaled, y = train_target,
    method = "knn", trControl = ctrl, tuneGrid = k_grid, metric = "ROC")

# best k
best_k <- knn_fit$bestTune$k
print(paste("Best k:", best_k))
```

```
[1] "Best k: 35"
```

```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  686 121
       Yes   0   0

               Accuracy : 0.8501
                 95% CI : (0.8235, 0.874)
    No Information Rate : 0.8501
    P-Value [Acc > NIR] : 0.5242

                  Kappa : 0

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.0000
            Specificity : 1.0000
         Pos Pred Value :    NaN
         Neg Pred Value : 0.8501
             Prevalence : 0.1499
         Detection Rate : 0.0000
   Detection Prevalence : 0.0000
      Balanced Accuracy : 0.5000

       'Positive' Class : Yes
```

Conclusion

The threshold optimization approach represents a clinically sound decision. In medical diagnostics, particularly for conditions like CHD, false negatives are typically more problematic than false positives. The selection of a lower threshold (0.14) increases sensitivity at the expense of some specificity. This trade-off is appropriate given: 1)The consequences of missing a CHD diagnosis could be severe 2)False positives typically lead to additional testing rather than immediate invasive treatment 3)The class imbalance requires special consideration to avoid a biased model Limitations and Recommendations: The kNN model appears to be severely affected by class imbalance. Implementation of class balancing techniques could improve its performance on minority class predictions. While logistic regression performs better at identifying positive cases, its negative predictive value remains low (26.01%). The analysis demonstrates a methodologically sound approach to CHD prediction, with appropriate consideration for the medical context in threshold selection. The logistic regression model provides more balanced predictions and is likely the more clinically useful model despite lower overall accuracy. Further optimization remains possible, particularly through addressing the class imbalance challenge.