A Lightweight Predictive Model for Heart Disease Based on Routine Clinical Data

Hanyu Wang

1 Introduction

Heart disease sometimes goes unnoticed, especially in asymptomatic individuals. However, occult heart disease can still cause shock and fatal consequences. Currently, diagnosing the disease requires special tests that are expensive and time-consuming. The aim of this study is to use low-cost, readily available physical data to predict the presence of heart disease.

Since liver disease, certain brain diseases and heart disease have similar characteristics, understanding statistical methods for predicting them can help predict heart disease. In previous studies, logistic regression was used in the prediction of liver disease (Abdalrada et al., 2019) and Alzheimer's disease (Johnson et al., 2014), and stepwise variable selection based on AIC was used in that of Alzheimer's disease (Johnson et al., 2014). In addition, Ture et al. (2008) used five models to predict the presence of coronary heart disease and suggested logistic regression to be a better performing technique. These suggest that it might be effective to use these methods to predict heart disease. Although Ture's study was similar to this one, the variables they ultimately considered included family history of CAD, diabetes mellitus, hypercholesterolemia and other difficult-to-obtain data. To enhance the applicability of the model, I utilized more accessible data and expanded the range of predicted diseases to the full spectrum of heart disease.

2 Methods

2.1 Choice of Method

Since the response variable is a binary, indicating whether the person has heart disease, we can use a binary logistic regression model. A generalized linear model (GLM) is proposed:

$$log(\frac{\pi}{1-\pi}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_q x_q = X\beta$$

2.2 Variable Selection

Akaike information criterion (AIC) is an estimator that can evaluate the fitness of the models, where $AIC = - 2L(\hat{\theta}) + 2p$; $L(\hat{\theta})$ is the maximized log-likelihood value, and p is the number of parameters. Stepwise selection based on AIC starts with a full model and drops a variable if it decreases AIC until all predictors are statistically significant. This method is used not only for it had been found efficient in previous studies, but also considering that it suggests a model that is more favorable for prediction instead of one with fewer predictors.

2.3 Model Violations and Diagnostics

a. Assumptions of Binary Logistic Regression:

(i) Binary Response: The response variable is dichotomous (two possible responses).
(ii) Independence: The observations are independent of each other.
(iii) Linearity: The log of the odds ratio, $log(\frac{\pi}{1-\pi})$, is a linear function of x.

(iv) Sufficient Sample Size: Each independent variable requires at minimum of 10 cases with the lowest frequency of outcomes.

b. Diagnostics: Dfbeta plots and deviance residual plots are used to check model diagnostics. Plots of an ideal model should display a random scatter of points symmetrically distributed around the identity line without discernible patterns.

c. Validation: Calibration and ROC plots are used to evaluate model performance. The bias-corrected line in the calibration plot of an ideal model should be a straight line starting from (0, 0) to (1, 1), and AUC of the ROC curve should be large and close to 1.

3 Results

3.1 Data Description

The data set contained information on 918 individuals, including eleven physical data and their heart disease status. After cleaning up the missing data, there were 746 observations remaining. To enhance the robustness of the model, all 746 observations were used.

The EDA plots are shown in Figure 4 in the Appendix.

Table 1: Dataset Summary

Numerical Variables						
Variables	Min.	1st Qu.	Median	Mean	3rd Qu.	Max
Age	28.00	46.00	54.00	52.88	59.00	77.00
RestingBP	92	120	130	133	140	200
Cholesterol	85.0	207.2	237.0	244.6	275.0	603.0
MaxHR	69.0	122.0	140.0	140.2	160.0	202.0
Oldpeak	-0.1000	0.0000	0.5000	0.9016	1.5000	6.2000
Nominal Variables						
Sex	Female: 193			Male: 725		
ChestPainType	ASY: 496	ATA: 173		NAP: 203	TA: 46	
FastingBS	No: 704			Yes: 214		
RestingECG	LVH: 188		Normal: 552		ST: 178	
ExerciseAngina	No: 547			Yes: 371		
ST_Slope	Down: 63		Flat: 460		Up: 395	
HeartDisease	No: 410			Yes: 508		

3.2 Analysis Proces

(i) A logistic regression model including all predictors was fitted.
(ii) Stepwise selections based on AIC and BIC, and LASSO method were used to identify the relevant predictors.
(iii) Models were fitted based on the selected predictors from the previous step.
(iv) Compare the modeling results (the model with smaller AIC and has predictors with larger estimators, smaller p-value and VIF has better performance), refer to previous similar studies, and select the most appropriate model.

The summary of the final model is shown in Table 2.

Table 2: Summary of the Final Model (Selected by Stepwise Selection Based on AIC)

(AIC: 542.33)	Estimate	Std. Error	p-value	VIF
Intercept	-2.764717	1.373911	0.044189	/
Age	0.034637	0.012895	0.007230	1.087973
SexMale	1.721240	0.296365	6.33e-09	1.101012
ChestPainTypeATA	-0.131288	0.516333	0.799287	2.787430
ChestPainTypeNAP	0.070530	0.480299	0.883254	3.374990
ChestPainTypeASY	1.600773	0.460741	0.000512	4.243226
RestingBP	0.011170	0.006929	0.106961	1.084407
Cholesterol	0.003334	0.001908	0.080556	1.051047
ExerciseAngina	1.076158	0.248973	1.54e-05	1.146023
Oldpeak	0.343885	0.138275	0.012884	1.282333
ST_Slope	-1.805422	0.236034	2.03e-14	1.272491

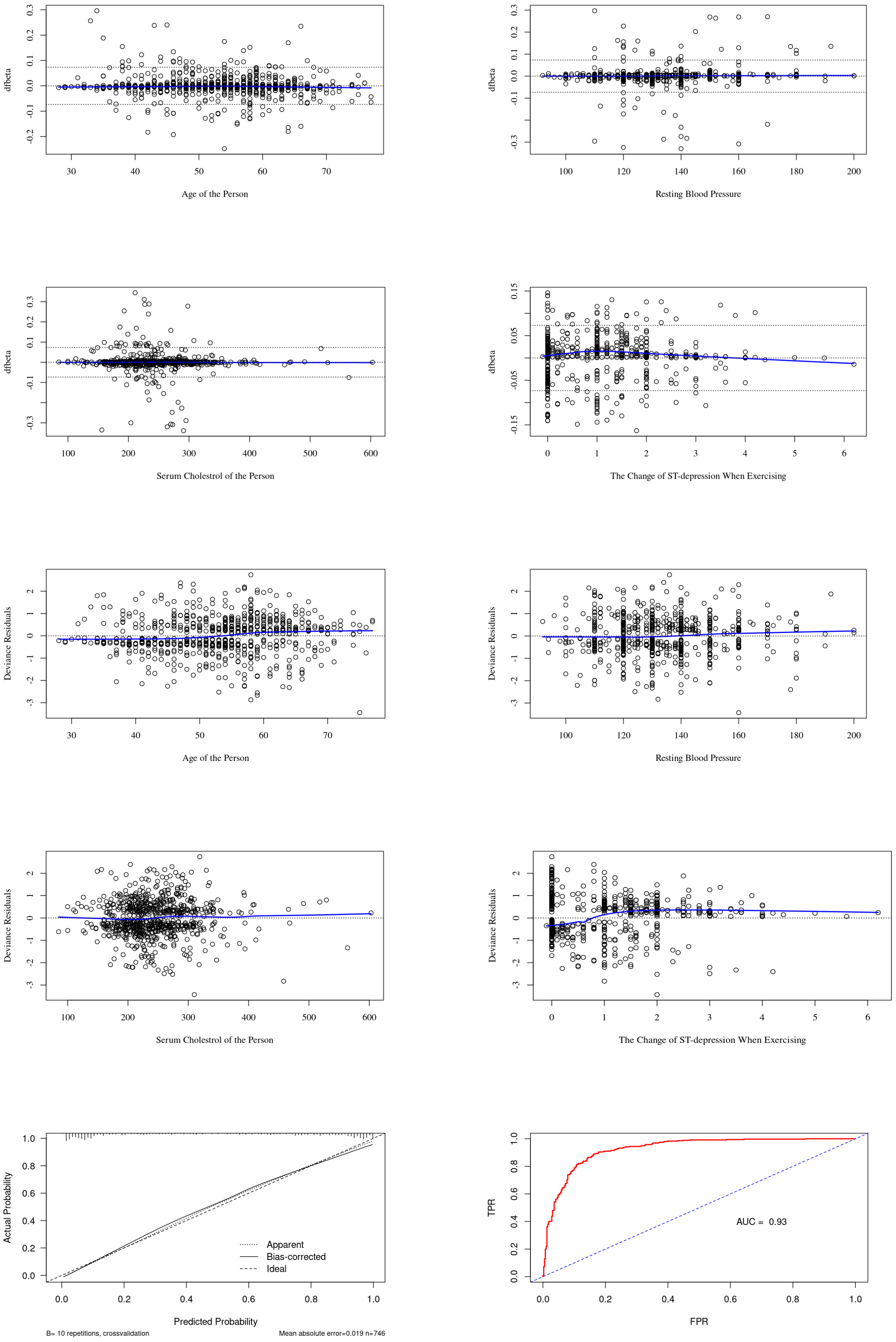
3.3 Diagnostics and Accuracy of the Final Model

The results of Dfbeta plots and deviance residual plots are mostly favorable, with most values very close to 0 and no clear pattern or trend. Although in the deviance residual plot of "The Change of ST-segment decline during exercise", there are some fluctuations in the residuals, there is no clear systematic pattern overall, meaning that the model does not show significant heteroskedasticity or nonlinear trends.

The bias-corrected line in the calibration plot is close to the ideal line. The AUC of the ROC curve is 0.93, indicating that the model can correctly distinguish 93% of the data.

These show the usability and good predictability of the model.

Figure 3: Final Model Diagnostics and Accuracy



4 Discussion

4.1 Final Model Interpretation and Importance

Since the estimated coefficient of Age is 0.034637, when it increases by 1, the odds ratio is $exp(0.034637) = 1.035$. Other numeric predictors can be interpreted in the same way.

For nominal variables, the predictor is 1 if that condition is met or 0 otherwise. For example, since the estimated coefficient of SexMale is 1.721240, holding other predictors constant, the odds ratio of male versus female is $exp(1.721240) = 5.591$, so males have larger probabilities to have heart disease than females.

To predict the presence of heart disease, if the predicted probability is larger than 0.5, the person is likely to have heart disease; otherwise, the person is more likely to be healthy.

The model has good performance, and all physical indicators needed are readily available, so this model solves the research question.

4.2 Limitations of the Analysis

The estimated coefficients of several predictors are small, with p-values larger than 0.05, indicating the predictors have small contributions to model prediction. However, since a model with accurate predictability is expected, this model was ultimately chosen considering that a model with a smaller AIC is more effective for prediction and previous similar studies have used AIC for model selection.

However, a model with most effective predictors was also fitted based on the results of stepwise selections based on AIC and BIC, and LASSO method. The details are shown in Table 5 in the Appendix for reference.

5 References

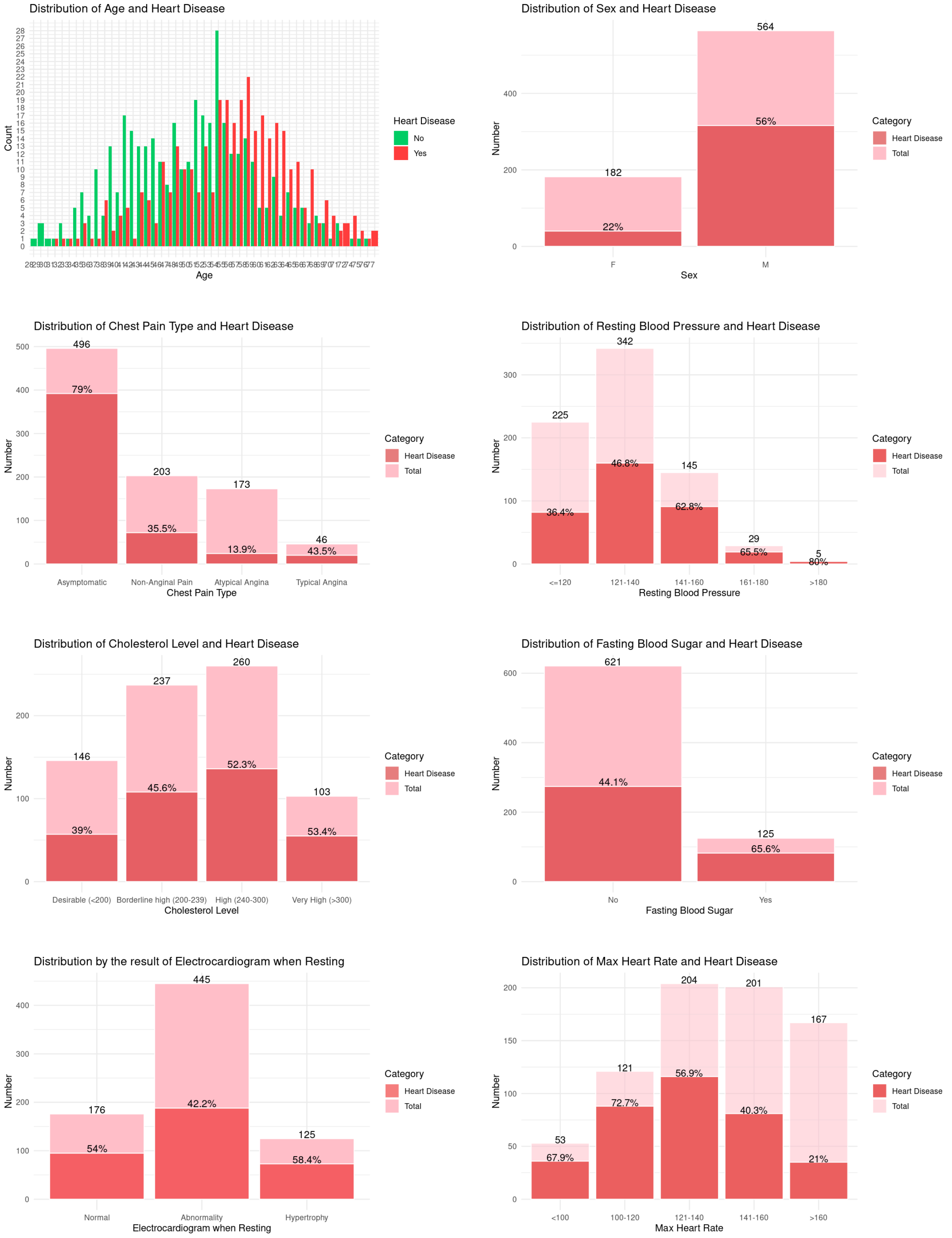
Johnson, P., Vandewater, L., Wilson, W., Maruff, P., Savage, G., Graham, P., Macaulay, L. S., Ellis, K. A., Szoeke, C., Martins, R. N., Rowe, C. C., Masters, C. L., Ames, D., & Zhang, P. (2014). Genetic algorithm with logistic regression for prediction of progression to Alzheimer's disease. BMC Bioinformatics, 15(Suppl 16), S11.

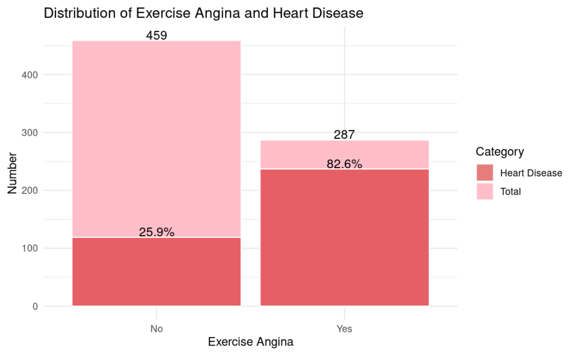
Kurt, I., Ture, M., & Kurum, A. T. (2008). Comparing performances of logistic regression, classification and regression tree, and neural networks for predicting coronary artery disease. Expert Systems with Applications, 34(1), 366–374.

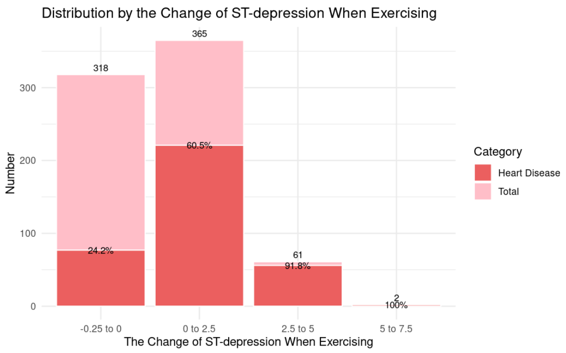
Abdalrada, A., Yahya, O., Alaidi, A., Hussein, N., Alrikabi, H., & Al-Quraishi, T.. (2019). A predictive model for liver disease progression based on logistic regression algorithm. Periodicals of Engineering and Natural Sciences (PEN), 7(3), 1255.

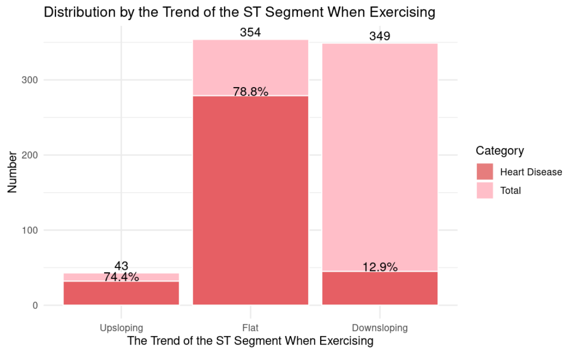
6 Appendix

Figure 4: EDA plots









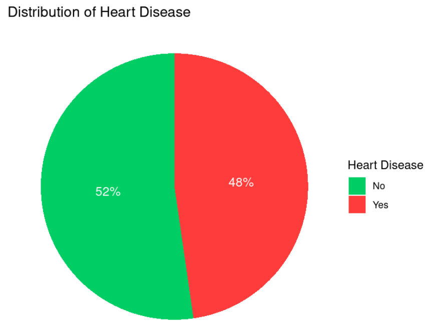


Table 5: Summary of the Fitted Model

(AIC: 559.15)	Estimate	Std. Error	p-value	VIF
Intercept	2.6749	0.6378	2.74e-05	/
SexMale	1.5528	0.2818	3.59e-08	1.053973
ChestPainTypeATA	-0.4343	0.4981	0.38323	2.709254
ChestPainTypeNAP	-0.1488	0.4642	0.74857	3.315481
ChestPainTypeASY	1.3918	0.4425	0.00166	4.111303
ExerciseAngina	1.3364	0.2361	1.51e-08	1.080341
ST_Slope	-2.1167	0.2151	< 2e-16	1.111114

7 Code

Hanyu Wang

Install Data & Packages

```
hd_initial=read.csv("/Users/macbook/Downloads/chd_data.csv")
```

Data Cleaning

```
hd <- hd_initial

hd <- hd %>%
  filter(RestingBP != 0, Cholesterol != 0)

hd <- hd %>%
  mutate(Sex = case_when(
    Sex == "F" ~ 0,
    Sex == "M" ~ 1),
    ChestPainType = case_when(
      ChestPainType == "TA" ~ 1,
      ChestPainType == "ATA" ~ 2,
      ChestPainType == "NAP" ~ 3,
      ChestPainType == "ASY" ~ 4),
    RestingECG = case_when(
      RestingECG == "Normal" ~ 1,
      RestingECG == "ST" ~ 2,
      RestingECG == "LVH" ~ 3),
    ExerciseAngina = case_when(
      ExerciseAngina == "N" ~ 0,
      ExerciseAngina == "Y" ~ 1),
    ST_Slope = case_when(
      ST_Slope == "Down" ~ 1,
      ST_Slope == "Flat" ~ 2,
      ST_Slope == "Up" ~ 3)
    )
```

Data Informations

Description

```
Variable <- c("Age (Numerical)", "Sex (Nominal)", "ChestPainType (Nominal)", "RestingBP (Numerical)", "

Description <- c("The age of the person (in year)",
                 "The sex of the person (0: F; 1: M)",
                 "The type of chest pain the person has (1: TA; 2: ATA; 3:NAP; 4: ASY)",
                 "The level of blood presure when the person is resting (in mm/HG)",
                 "Whether the Blood sugear level on fasting of the person achieve 120 mg/dl (0: N; 1: Y
                 "The maximum heart rate of the person",
                 "The ST-depression when exercising compare to when resting",
                 "The slope of the ST segment when exercising (0: Normal; 1: Upsloping; 2: Flat; 3: Dow
                 "The Serum cholestrol of the person (in mg/dl)",
                 "The result of electrocardiogram when the person is resting (0: Normal; 1: ST; 2: LVH)
                 "Whether the person has angina when exercising (0: N; 1: Y)",
                 "Whether the person have the heart disease (0: False; 1: True)")

Type <- c("Predictor of Interest", "Predictor of Interest", "Predictor of Interest", "Predictor of Inte

info <- tibble(Variable, Description, Type)
table <- knitr::kable(info)
table
```

Variable	Description	Type
Age (Numerical)	The age of the person (in year)	Predictor of Interest
Sex (Nominal)	The sex of the person (0: F; 1: M)	Predictor of Interest
ChestPainType (Nominal)	The type of chest pain the person has (1: TA; 2: ATA; 3:NAP; 4: ASY)	Predictor of Interest
RestingBP (Numerical)	The level of blood presure when the person is resting (in mm/HG)	Predictor of Interest
FastingBS (Nominal)	Whether the Blood sugear level on fasting of the person achieve 120 mg/dl (0: N; 1: Y)	Predictor of Interest
MaxHR (Numerical)	The maximum heart rate of the person	Predictor of Interest
Oldpeak (Numerical)	The ST-depression when exercising compare to when resting	Predictor of Interest
ST_Slope (Nominal)	The slope of the ST segment when exercising (0: Normal; 1: Upsloping; 2: Flat; 3: Downsloping)	Predictor of Interest
Cholestrol (Numerical)	The Serum cholestrol of the person (in mg/dl)	Confounding Variabl
RestingECG (Nominal)	The result of electrocardiogram when the person is resting (0: Normal; 1: ST; 2: LVH)	Confounding Variabl
ExerciseAngina (Nominal)	Whether the person has angina when exercising (0: N; 1: Y)	Confounding Variable
HeartDisease (Nominal)	Whether the person have the heart disease (0: False; 1: True)	Response Variable

Summary

```
hd_initial <- hd_initial %>%
  filter(RestingBP != 0, Cholesterol != 0)

summary(hd_initial[c(1, 4, 5, 8, 10)])
```

```
##       Age          RestingBP     Cholesterol         MaxHR
##  Min.   :28.00   Min.   : 92   Min.   : 85.0   Min.   : 69.0
##  1st Qu.:46.00   1st Qu.:120   1st Qu.:207.2   1st Qu.:122.0
##  Median :54.00   Median :130   Median :237.0   Median :140.0
##  Mean   :52.88   Mean   :133   Mean   :244.6   Mean   :140.2
##  3rd Qu.:59.00   3rd Qu.:140   3rd Qu.:275.0   3rd Qu.:160.0
##  Max.   :77.00   Max.   :200   Max.   :603.0   Max.   :202.0
##     Oldpeak
##  Min.   :-0.1000
##  1st Qu.: 0.0000
##  Median : 0.5000
##  Mean   : 0.9016
##  3rd Qu.: 1.5000
##  Max.   : 6.2000
```

```
table(hd_initial$Sex)
```

```
##
##   F   M
## 182 564
```

```
table(hd_initial$ChestPainType)
```

```
##
## ASY ATA NAP  TA
## 370 166 169  41
```

```
table(hd_initial$FastingBS)
```

```
##
##   0   1
## 621 125
```

```
table(hd_initial$RestingECG)
```

```
##
##    LVH Normal     ST
##    176    445    125
```

```
table(hd_initial$ExerciseAngina)
```

```
##
##   N   Y
## 459 287
```

```
table(hd_initial$ST_Slope)
```

```
##
## Down Flat   Up
##   43  354  349
```

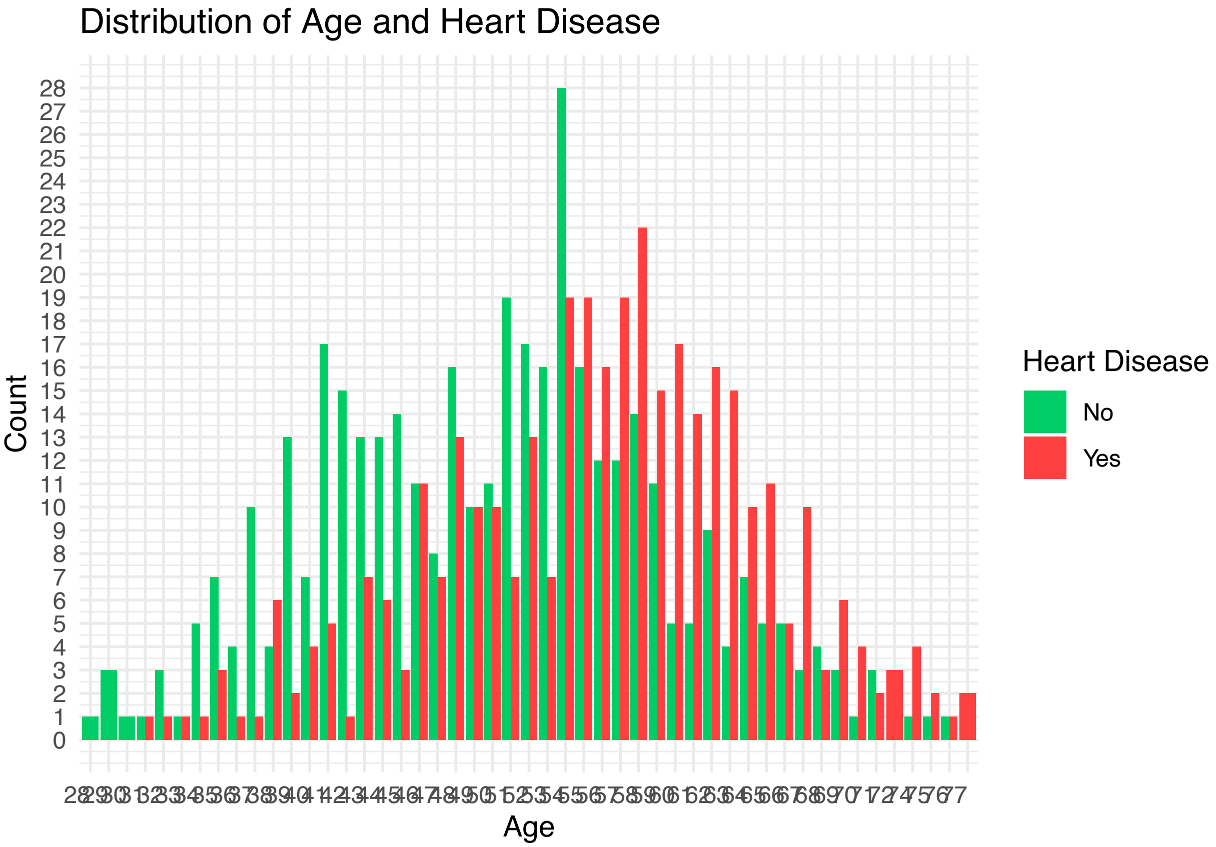
```
table(hd_initial$HeartDisease)
```

```
##
##   0   1
## 390 356
```

Distributions

Age

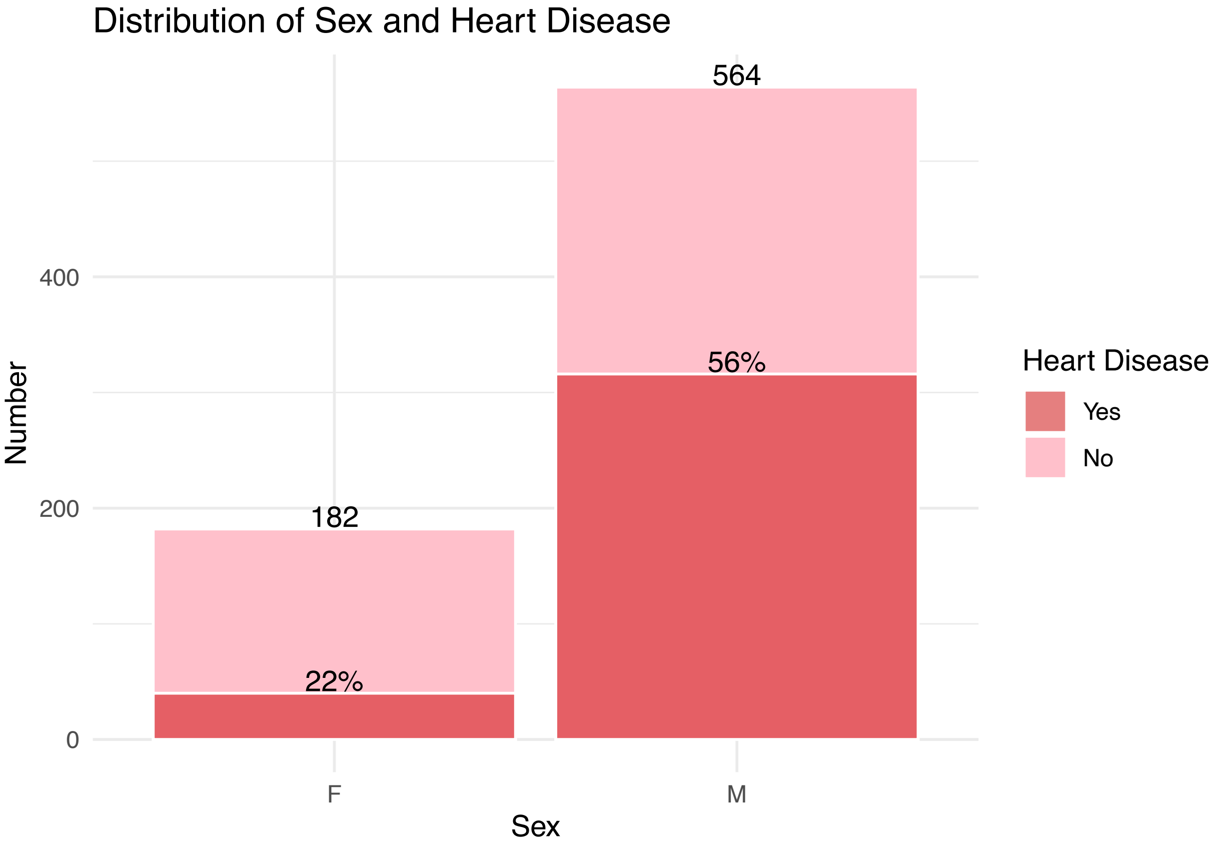
```
temp_hd <- hd %>%
  mutate(HeartDisease = factor(HeartDisease, levels = c(0, 1), labels = c("No", "Yes")))

ggplot(temp_hd, aes(x = as.factor(Age), fill = HeartDisease)) +
  geom_bar(position = "dodge") +
  scale_fill_manual(values = c("No" = "springgreen3", "Yes" = "brown1")) +
  scale_y_continuous(breaks = seq(0, max(table(temp_hd$Age, temp_hd$HeartDisease)), by = 1)) +
  labs(title = "Distribution of Age and Heart Disease",
       x = "Age",
       y = "Count",
       fill = "Heart Disease") +
  theme_minimal() +
  theme(axis.text.x = element_text(hjust = 1))
```



Sex

```
sex_counts <- hd_initial %>%
  count(Sex) %>%
  mutate(Percentage = n / sum(n) * 100)

heart_disease_counts <- hd_initial %>%
  filter(HeartDisease == 1) %>%
  count(Sex)
```

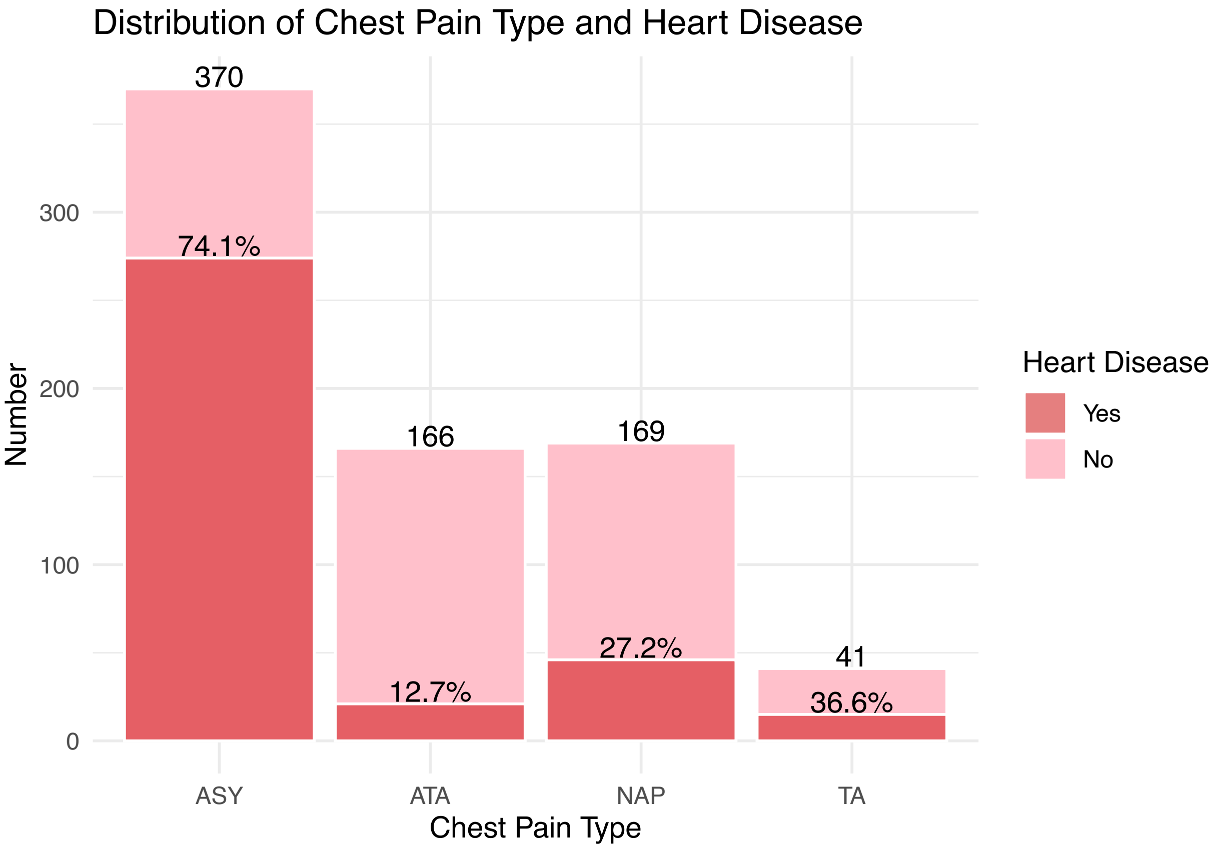
```
total_counts <- hd_initial %>%
  count(Sex)

heart_disease_prop <- heart_disease_counts %>%
  inner_join(total_counts, by = "Sex") %>%
  mutate(prop = n.x / n.y)

ggplot() +
  geom_bar(data = sex_counts, aes(x = factor(Sex), y = n, fill = "No Heart Disease"), stat = "identity"
  geom_text(data = sex_counts, aes(x = factor(Sex), label = n, y = n), vjust = -0.1, size = 4) +
  geom_bar(data = heart_disease_prop, aes(x = factor(Sex), y = prop * n.y, fill = "Heart Disease"), sta
  geom_text(data = heart_disease_prop, aes(x = factor(Sex), label = paste0(round(prop * 100, 1), "%"), y
  scale_fill_manual(values = c("red3", "pink"), name = "Heart Disease", labels = c("Yes", "No")) +
  labs(x = "Sex", y = "Number", fill = "Heart Disease", title = "Distribution of Sex and Heart Disease")
  theme_minimal()
```



ChestPainType

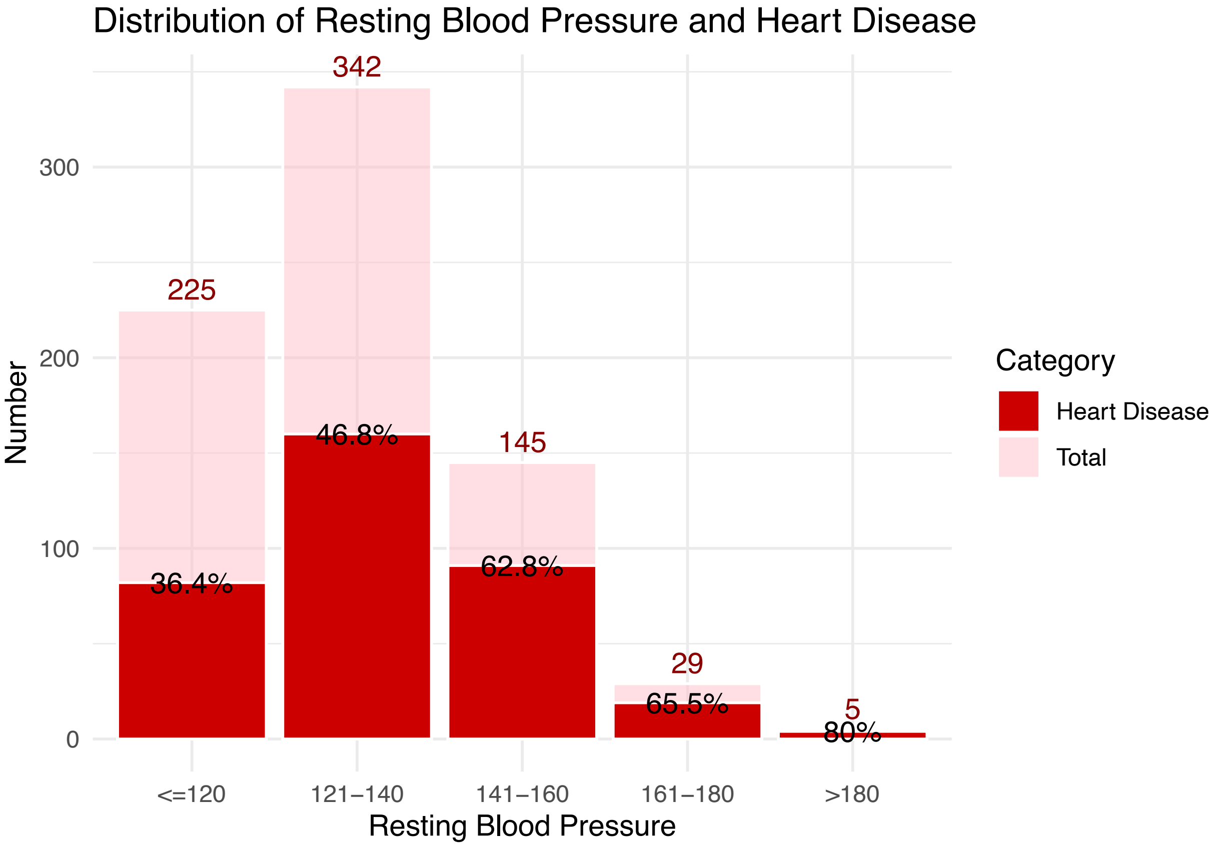
```
chest_pain_counts <- hd_initial %>%
  count(ChestPainType) %>%
  mutate(Percentage = n / sum(n) * 100)

heart_disease_counts <- hd_initial %>%
  filter(HeartDisease == 1) %>%
  count(ChestPainType)
```

```
total_counts <- hd_initial %>%
  count(ChestPainType)

heart_disease_prop <- heart_disease_counts %>%
  inner_join(total_counts, by = "ChestPainType") %>%
  mutate(prop = n.x / n.y)

ggplot() +
  geom_bar(data = chest_pain_counts, aes(x = factor(ChestPainType), y = n, fill = "No Heart Disease"), s
  geom_text(data = chest_pain_counts, aes(x = factor(ChestPainType), label = n, y = n), vjust = -0.1, si
  geom_bar(data = heart_disease_prop, aes(x = factor(ChestPainType), y = prop * n.y, fill = "Heart Disea
  geom_text(data = heart_disease_prop, aes(x = factor(ChestPainType), label = paste0(round(prop * 100, 1
  scale_fill_manual(values = c("red3", "pink"), name = "Heart Disease", labels = c("Yes", "No")) +
  labs(x = "Chest Pain Type", y = "Number", fill = "Heart Disease", title = "Distribution of Chest Pain
  theme_minimal()
```



Resting BP

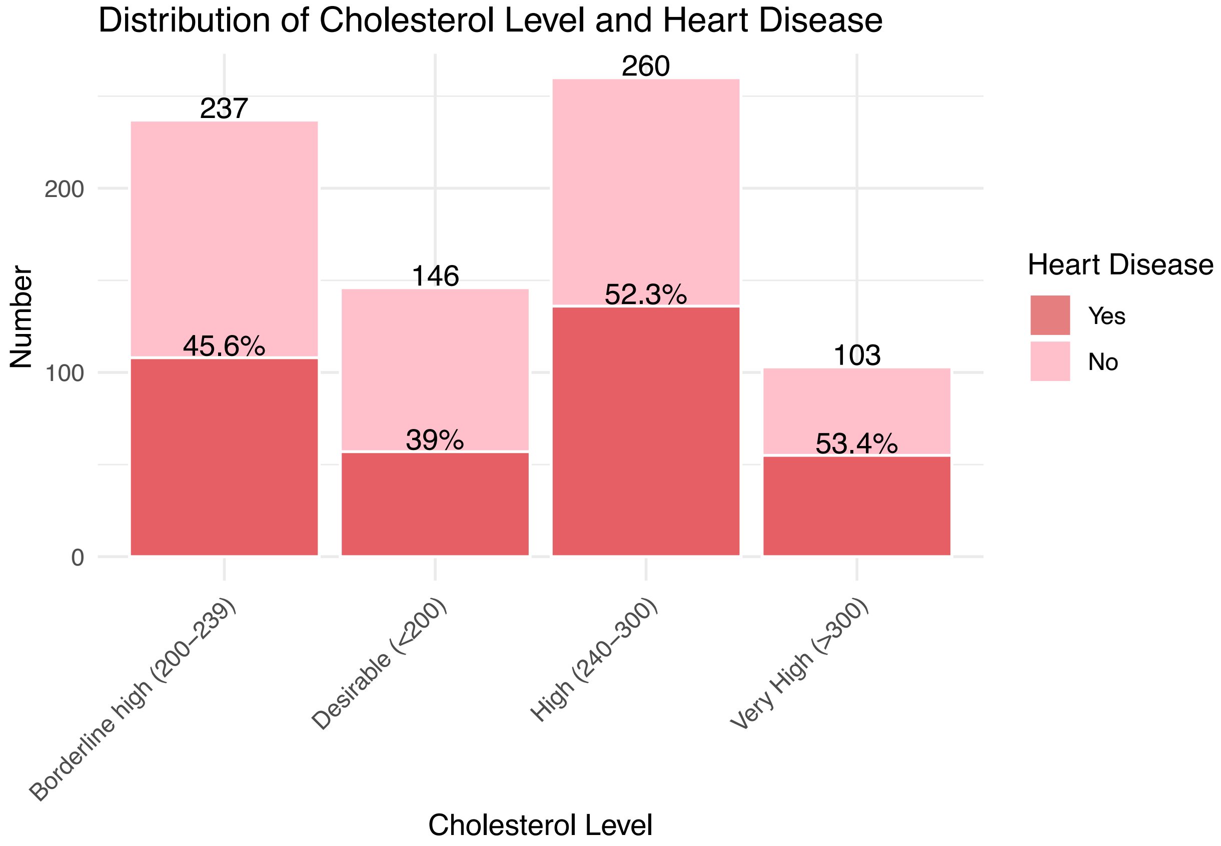
```
hd$RestingBPGroup <- cut(hd$RestingBP,
                          breaks = c(0, 120, 140, 160, 180, Inf),
                          labels = c("<=120", "121-140", "141-160", "161-180", ">180"),
                          include.lowest = TRUE)

bp_counts <- hd %>%
  group_by(RestingBPGroup) %>%
  summarise(Total = n(),
            HeartDiseaseCount = sum(HeartDisease == 1),
            Percentage = HeartDiseaseCount / Total * 100)

ggplot(bp_counts, aes(x = RestingBPGroup)) +
  geom_bar(aes(y = Total, fill = "Total"), stat = "identity", color = "white", alpha = 0.5) +
  geom_bar(aes(y = HeartDiseaseCount, fill = "Heart Disease"), stat = "identity", color = "white") +
  geom_text(aes(y = Total, label = Total), vjust = -0.5, color = "darkred") +
  geom_text(aes(y = HeartDiseaseCount, label = paste0(round(Percentage, 1), "%")), vjust = +0.5, color =
  scale_fill_manual(values = c("Total" = "pink", "Heart Disease" = "red3"), name = "Category") +
  labs(title = "Distribution of Resting Blood Pressure and Heart Disease",
       x = "Resting Blood Pressure", y = "Number") +
  theme_minimal()
```



```
hd <- subset(hd, select = -RestingBPGroup)
```

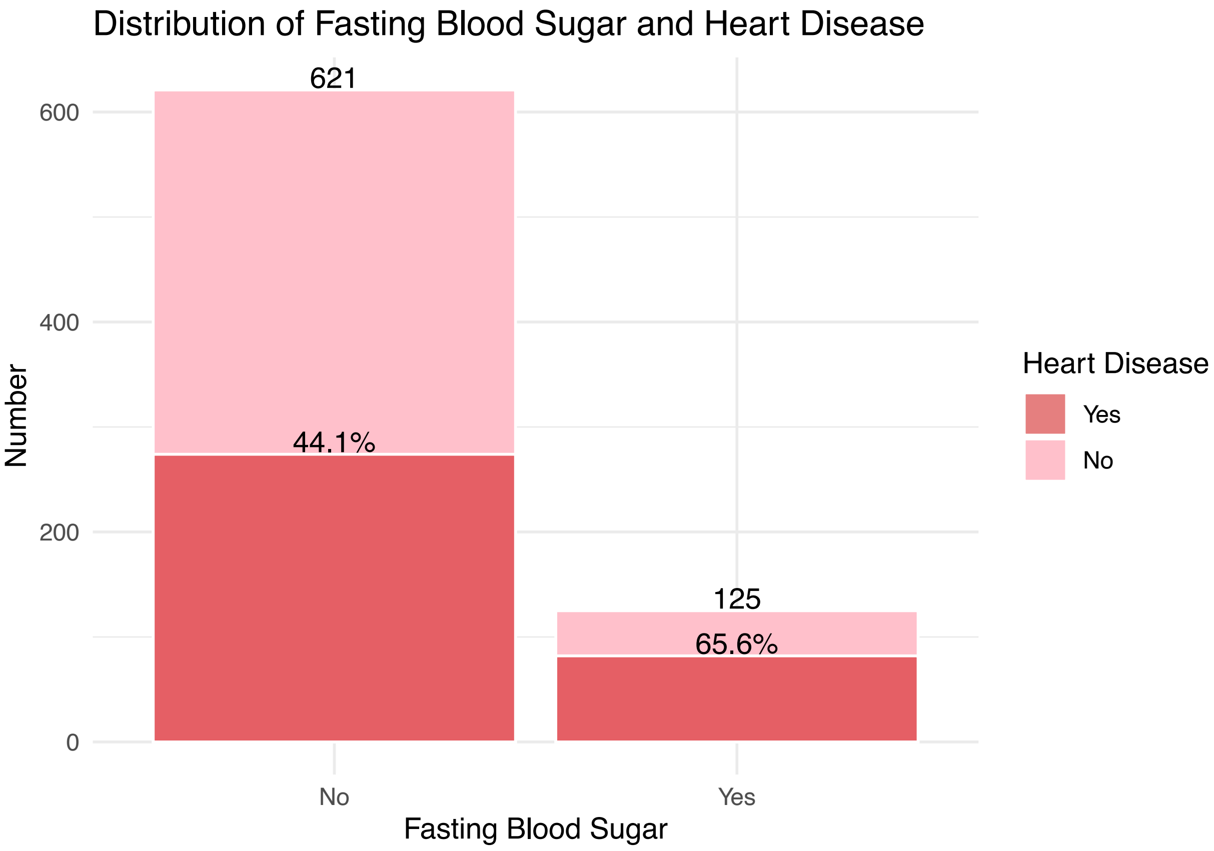
Cholesterol

```
hd_initial <- hd_initial %>%
  mutate(Cholesterol_Group = case_when(
    Cholesterol < 200 ~ "Desirable (<200)",
    Cholesterol >= 200 & Cholesterol <= 239 ~ "Borderline high (200-239)",
    Cholesterol > 239 & Cholesterol <= 300 ~ "High (240-300)",
    Cholesterol > 300 ~ "Very High (>300)"
  ))

cholesterol_counts <- hd_initial %>%
  count(Cholesterol_Group) %>%
  mutate(Percentage = n / sum(n) * 100)

heart_disease_counts <- hd_initial %>%
  filter(HeartDisease == 1) %>%
  count(Cholesterol_Group)

total_counts <- hd_initial %>%
  count(Cholesterol_Group)

heart_disease_prop <- heart_disease_counts %>%
  inner_join(total_counts, by = "Cholesterol_Group") %>%
  mutate(prop = n.x / n.y)

ggplot() +
  geom_bar(data = cholesterol_counts, aes(x = Cholesterol_Group, y = n, fill = "No Heart Disease"), stat
  geom_text(data = cholesterol_counts, aes(x = Cholesterol_Group, label = n, y = n), vjust = -0.1, size
  geom_bar(data = heart_disease_prop, aes(x = Cholesterol_Group, y = prop * n.y, fill = "Heart Disease")
  geom_text(data = heart_disease_prop, aes(x = Cholesterol_Group, label = paste0(round(prop * 100, 1),
  scale_fill_manual(values = c("red3", "pink"), name = "Heart Disease", labels = c("Yes", "No")) +
  labs(x = "Cholesterol Level", y = "Number", fill = "Heart Disease", title = "Distribution of Choleste
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



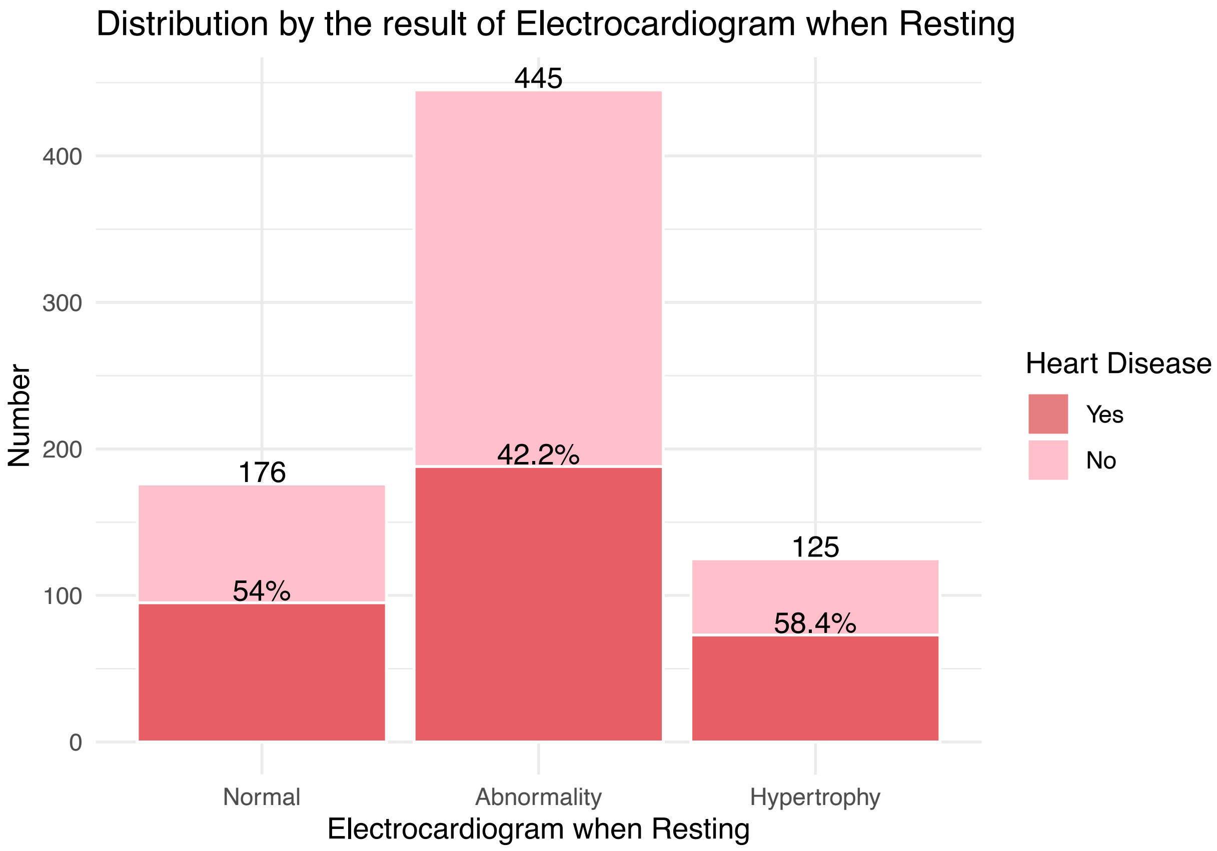
FastingBS

```
fasting_bs_counts <- hd_initial %>%
  count(FastingBS) %>%
  mutate(Percentage = n / sum(n) * 100)

heart_disease_counts <- hd_initial %>%
  filter(HeartDisease == 1) %>%
  count(FastingBS)

total_counts <- hd_initial %>%
  count(FastingBS)

heart_disease_prop <- heart_disease_counts %>%
  inner_join(total_counts, by = "FastingBS") %>%
  mutate(prop = n.x / n.y)

ggplot() +
  geom_bar(data = fasting_bs_counts, aes(x = factor(FastingBS, labels = c("No", "Yes")), y = n, fill =
  geom_text(data = fasting_bs_counts, aes(x = factor(FastingBS, labels = c("No", "Yes")), label = n, y =
  geom_bar(data = heart_disease_prop, aes(x = factor(FastingBS, labels = c("No", "Yes")), y = prop * n.y
  geom_text(data = heart_disease_prop, aes(x = factor(FastingBS, labels = c("No", "Yes")), label = paste
  scale_fill_manual(values = c("red3", "pink"), name = "Heart Disease", labels = c("Yes", "No")) +
  labs(x = "Fasting Blood Sugar", y = "Number", fill = "Heart Disease", title = "Distribution of Fasting
  theme_minimal()
```



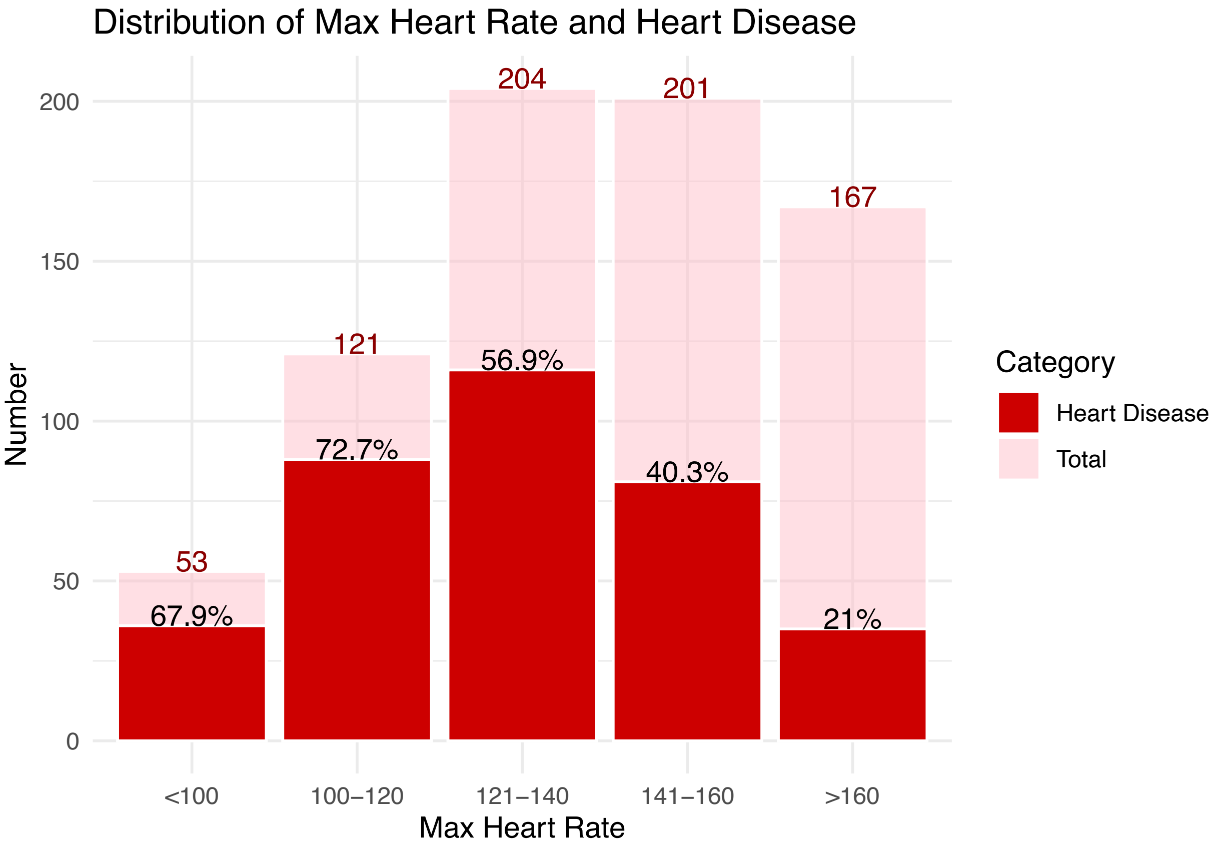
RestingECG

```
resting_ecg_counts <- hd_initial %>%
  count(RestingECG) %>%
  mutate(Percentage = n / sum(n) * 100)

heart_disease_counts <- hd_initial %>%
  filter(HeartDisease == 1) %>%
  count(RestingECG)

total_counts <- hd_initial %>%
  count(RestingECG)

heart_disease_prop <- heart_disease_counts %>%
  inner_join(total_counts, by = "RestingECG") %>%
  mutate(prop = n.x / n.y)

ggplot() +
  geom_bar(data = resting_ecg_counts, aes(x = factor(RestingECG, labels = c("Normal", "Abnormality", "Hy
  geom_text(data = resting_ecg_counts, aes(x = factor(RestingECG, labels = c("Normal", "Abnormality", "
  geom_bar(data = heart_disease_prop, aes(x = factor(RestingECG, labels = c("Normal", "Abnormality", "Hy
  geom_text(data = heart_disease_prop, aes(x = factor(RestingECG, labels = c("Normal", "Abnormality", "
  scale_fill_manual(values = c("red3", "pink"), name = "Heart Disease", labels = c("Yes", "No")) +
  labs(x = "Electrocardiogram when Resting", y = "Number", fill = "Heart Disease", title = "Distribution
  theme_minimal()
```



MaxHR

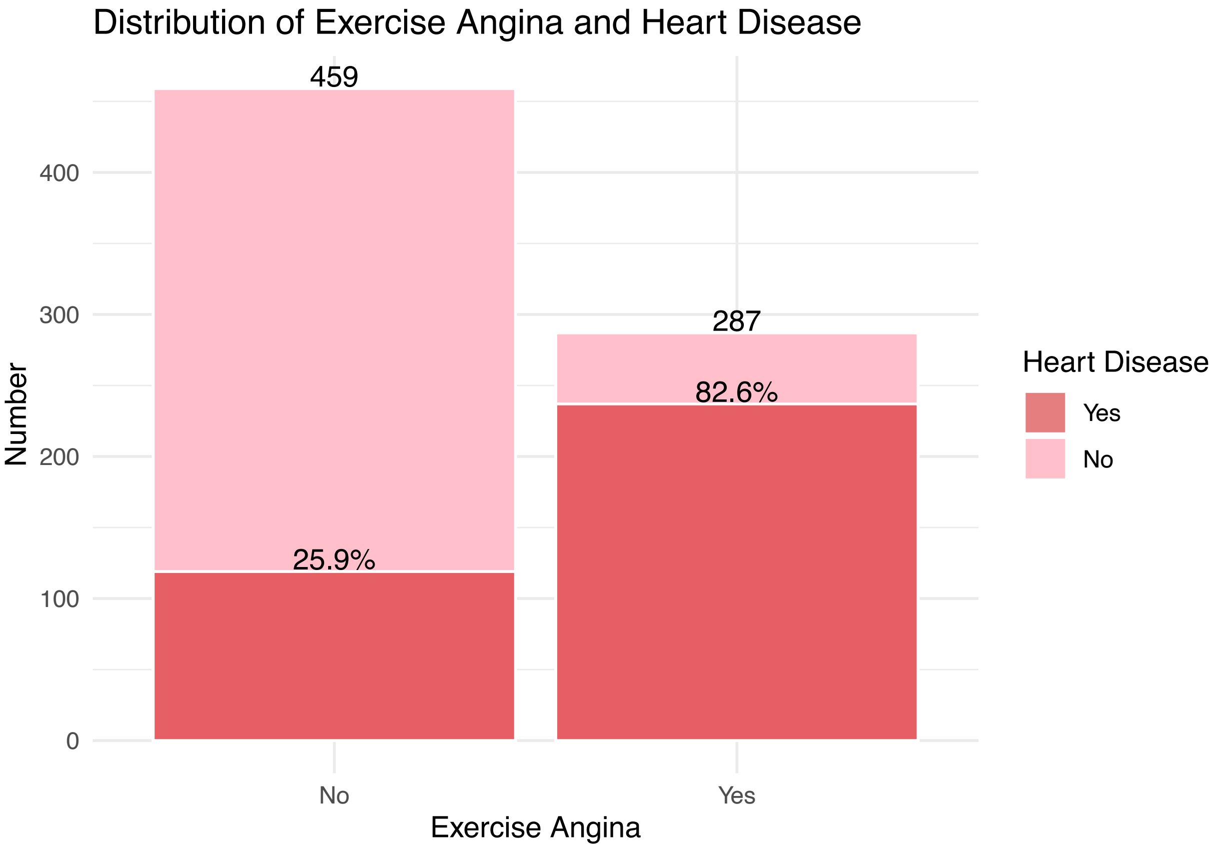
```r
maxhr_counts <- hd %>%
  filter(MaxHR != 0) %>%
  mutate(MaxHR_Group = cut(MaxHR, breaks = c(0, 100, 120, 140, 160, Inf),
                           labels = c("<100", "100-120", "121-140", "141-160", ">160"),
                           include.lowest = TRUE)) %>%
  group_by(MaxHR_Group) %>%
  summarise(Total = n(),
            HeartDiseaseCount = sum(HeartDisease == 1),
            Percentage = HeartDiseaseCount / Total * 100)

ggplot(maxhr_counts, aes(x = MaxHR_Group)) +
  geom_bar(aes(y = Total, fill = "Total"), stat = "identity", color = "white", alpha = 0.5) +
  geom_bar(aes(y = HeartDiseaseCount, fill = "Heart Disease"), stat = "identity", color = "white") +
  geom_text(aes(y = Total, label = Total), vjust = 0, color = "darkred") +
  geom_text(aes(y = HeartDiseaseCount, label = paste0(round(Percentage, 1), "%")), vjust = 0, color = "
  scale_fill_manual(values = c("Total" = "pink", "Heart Disease" = "red3"), name = "Category") +
  labs(title = "Distribution of Max Heart Rate and Heart Disease",
       x = "Max Heart Rate", y = "Number") +
  theme_minimal()
```



ExerciseAngina

```r
exercise_angina_counts <- hd_initial %>%
  count(ExerciseAngina) %>%
  mutate(Percentage = n / sum(n) * 100)
```

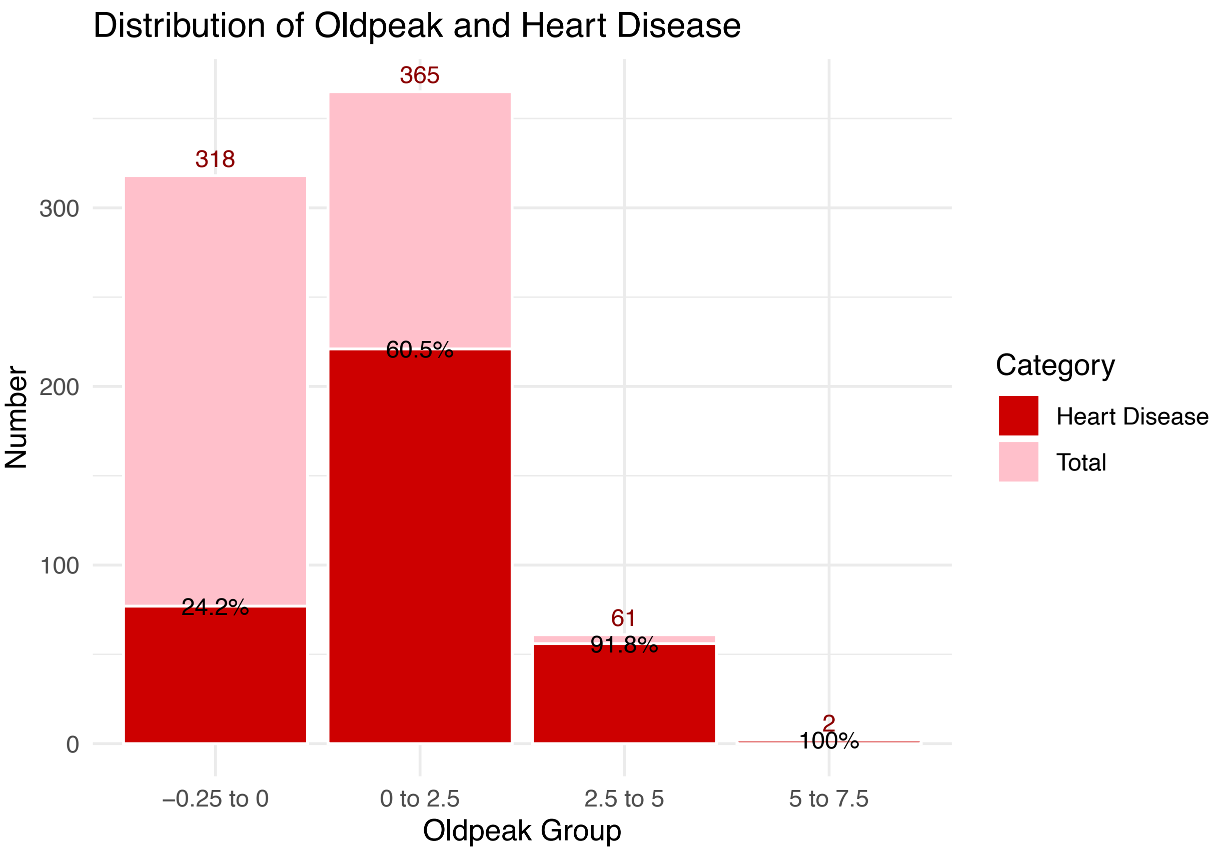
```r
heart_disease_counts <- hd_initial %>%
  filter(HeartDisease == 1) %>%
  count(ExerciseAngina)

total_counts <- hd_initial %>%
  count(ExerciseAngina)

heart_disease_prop <- heart_disease_counts %>%
  inner_join(total_counts, by = "ExerciseAngina") %>%
  mutate(prop = n.x / n.y)

ggplot() +
  geom_bar(data = exercise_angina_counts, aes(x = factor(ExerciseAngina, labels = c("No", "Yes")), y = n
  geom_text(data = exercise_angina_counts, aes(x = factor(ExerciseAngina, labels = c("No", "Yes")), labe
  geom_bar(data = heart_disease_prop, aes(x = factor(ExerciseAngina, labels = c("No", "Yes")), y = prop
  geom_text(data = heart_disease_prop, aes(x = factor(ExerciseAngina, labels = c("No", "Yes")), label =
  scale_fill_manual(values = c("red3", "pink"), name = "Heart Disease", labels = c("Yes", "No")) +
  labs(x = "Exercise Angina", y = "Number", fill = "Heart Disease", title = "Distribution of Exercise An
  theme_minimal()
```



Oldpeak

```r
hd$Oldpeak_Group <- cut(hd$Oldpeak,
                        breaks = c(-5, -0.25, 0, 2.5, 5, 7.5),
                        labels = c("-5 to -0.25", "-0.25 to 0", "0 to 2.5", "2.5 to 5", "5 to 7.5"),
                        include.lowest = TRUE)
```

```
oldpeak_counts <- hd %>%
  group_by(Oldpeak_Group) %>%
  summarise(Total = n(),
            HeartDiseaseCount = sum(HeartDisease == 1))

ggplot(oldpeak_counts, aes(x = Oldpeak_Group)) +
  geom_bar(aes(y = Total, fill = "Total"), stat = "identity", color = "white") +
  geom_bar(aes(y = HeartDiseaseCount, fill = "Heart Disease"), stat = "identity", color = "white") +
  geom_text(aes(y = Total, label = Total), vjust = -0.5, color = "darkred", size = 3) +
  geom_text(aes(y = HeartDiseaseCount, label = paste0(round(HeartDiseaseCount / Total * 100, 1), "%")),
  labs(title = "Distribution of Oldpeak and Heart Disease",
       x = "Oldpeak Group", y = "Number") +
  scale_fill_manual(values = c("Total" = "pink", "Heart Disease" = "red3"), name = "Category") +
  theme_minimal()
```

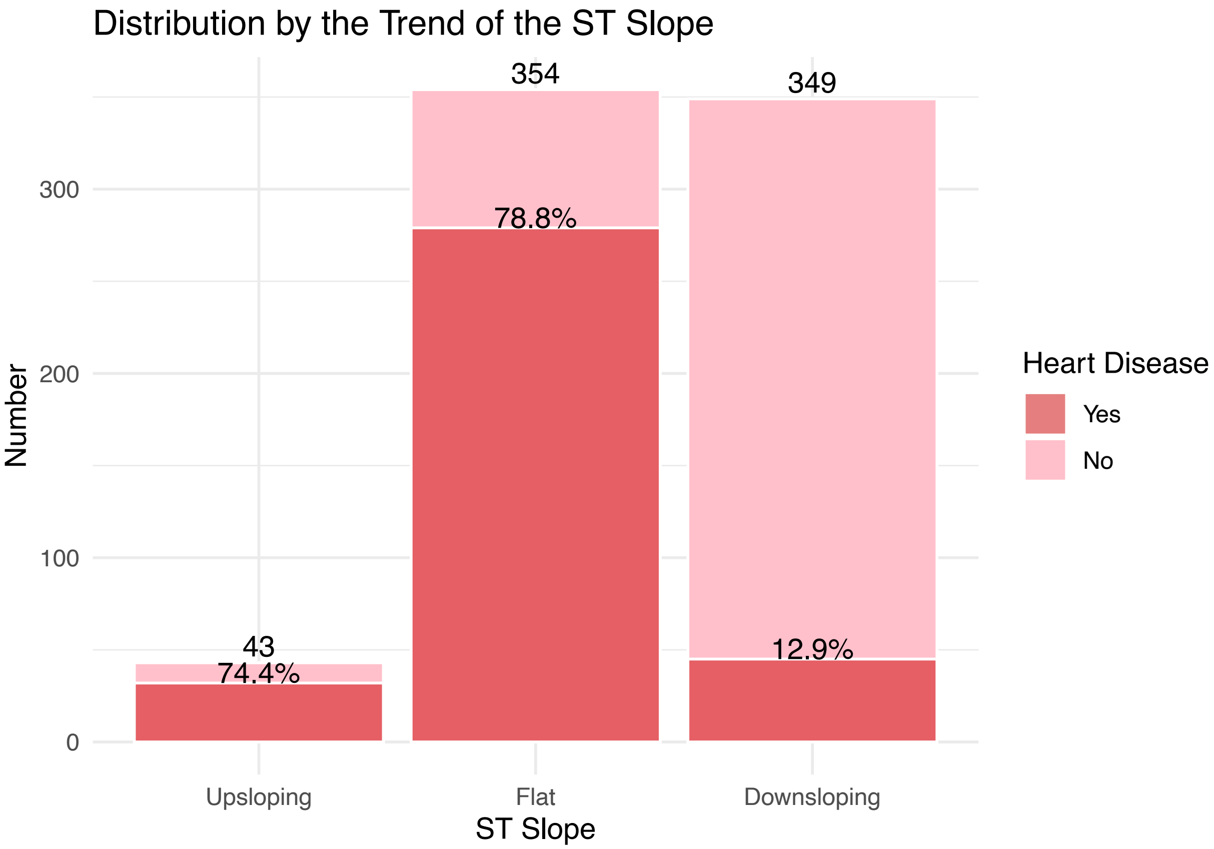


```
hd <- subset(hd, select = -Oldpeak_Group)
```

ST_Slope

```
st_slope_counts <- hd_initial %>%
  count(ST_Slope) %>%
  mutate(Percentage = n / sum(n) * 100)

heart_disease_counts <- hd_initial %>%
  filter(HeartDisease == 1) %>%
  count(ST_Slope)
```

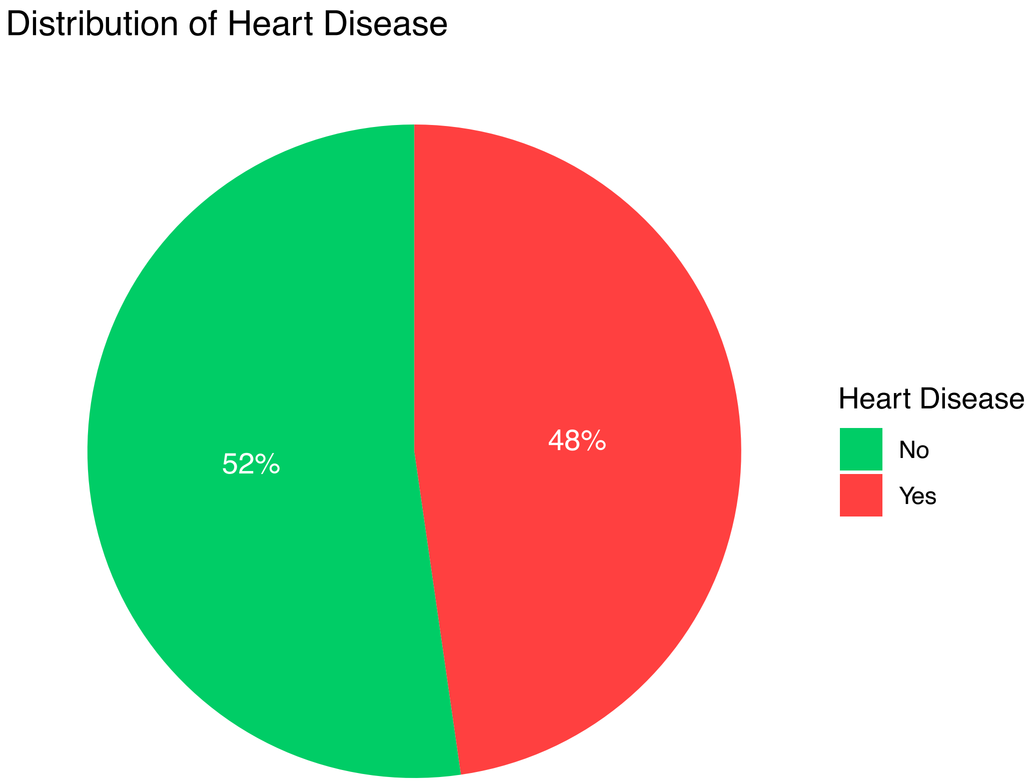
```
total_counts <- hd_initial %>%
  count(ST_Slope)

heart_disease_prop <- heart_disease_counts %>%
  inner_join(total_counts, by = "ST_Slope") %>%
  mutate(prop = n.x / n.y)

ggplot() +
  geom_bar(data = st_slope_counts, aes(x = factor(ST_Slope, labels = c("Upsloping", "Flat", "Downsloping
  geom_text(data = st_slope_counts, aes(x = factor(ST_Slope, labels = c("Upsloping", "Flat", "Downslopin
  geom_bar(data = heart_disease_prop, aes(x = factor(ST_Slope, labels = c("Upsloping", "Flat", "Downslop
  geom_text(data = heart_disease_prop, aes(x = factor(ST_Slope, labels = c("Upsloping", "Flat", "Downsl
  scale_fill_manual(values = c("red3", "pink"), name = "Heart Disease", labels = c("Yes", "No")) +
  labs(x = "ST Slope", y = "Number", fill = "Heart Disease", title = "Distribution by the Trend of the
  theme_minimal()
```



HeartDisease

```
heartdisease_summary <- hd %>%
  count(HeartDisease)

total <- sum(heartdisease_summary$n)

heartdisease_summary <- heartdisease_summary %>%
  mutate(percentage = n / total * 100)
```

```
ggplot(heartdisease_summary, aes(x = "", y = n, fill = factor(HeartDisease))) +
  geom_bar(stat = "identity", width = 1) +
  geom_text(aes(label = paste0(round(percentage), "%")),
            position = position_stack(vjust = 0.5), color = "white", size = 4) +
  coord_polar(theta = "y") +
  labs(title = "Distribution of Heart Disease", fill = "Heart Disease") +
  scale_fill_manual(values = c("0" = "springgreen3", "1" = "brown1"), labels = c("No", "Yes")) +
  theme_void()
```



Model

Full Model

```
FullModel <- glm(as.factor(HeartDisease) ~ ., data=hd, family="binomial")
summary(FullModel)
```

```
##
## Call:
## glm(formula = as.factor(HeartDisease) ~ ., family = "binomial",
##     data = hd)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -3.629399   1.647312  -2.203  0.02758 *
## Age               0.028085   0.013888   2.022  0.04315 *
## Sex               1.715656   0.290171   5.913 3.37e-09 ***
## ChestPainType     0.721247   0.126778   5.689 1.28e-08 ***
## RestingBP         0.010982   0.006877   1.597  0.11025
## Cholesterol     0.003529   0.001915   1.843  0.06536 .
## FastingBS       0.232078   0.310404   0.748  0.45466
## RestingECG      0.085777   0.133609   0.642  0.52087
## MaxHR          -0.004319   0.005243  -0.824  0.41005
## ExerciseAngina  1.114167   0.250737   4.444 8.85e-06 ***
## Oldpeak         0.370779   0.136599   2.714  0.00664 **
## ST_Slope       -1.765163   0.237584  -7.430 1.09e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1032.63  on 745  degrees of freedom
## Residual deviance:  531.01  on 734  degrees of freedom
## AIC: 555.01
##
## Number of Fisher Scoring iterations: 5
```

Select Predictors

Stepwise Selection Base on AIC

```
sel.var.aic <- step(FullModel, trace = 0, k = 2, direction = "both")
select_var_aic<-attr(terms(sel.var.aic), "term.labels")
select_var_aic
```

```
## [1] "Age"            "Sex"            "ChestPainType"  "RestingBP"
## [5] "Cholesterol"    "ExerciseAngina" "Oldpeak"        "ST_Slope"
```

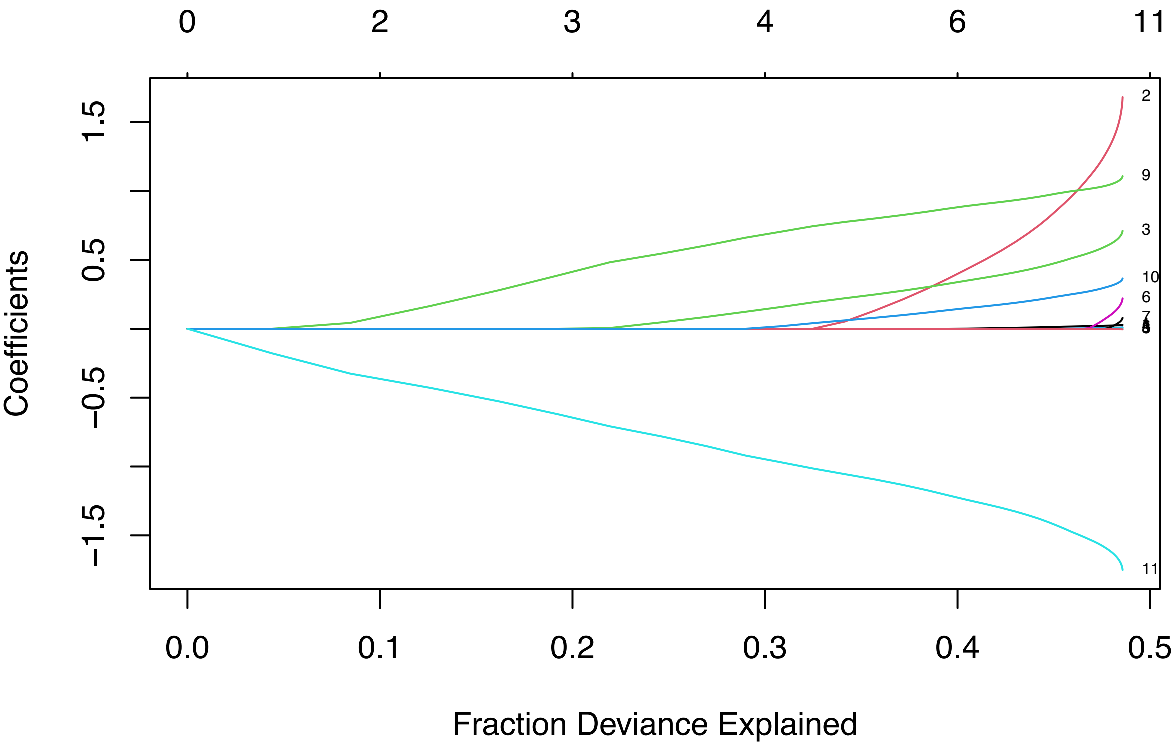
Stepwise Selection Base on BIC

```
sel.var.bic <- step(FullModel, trace = 0, k = log(nrow(hd)), direction = "both")
select_var_bic<-attr(terms(sel.var.bic), "term.labels")
select_var_bic
```

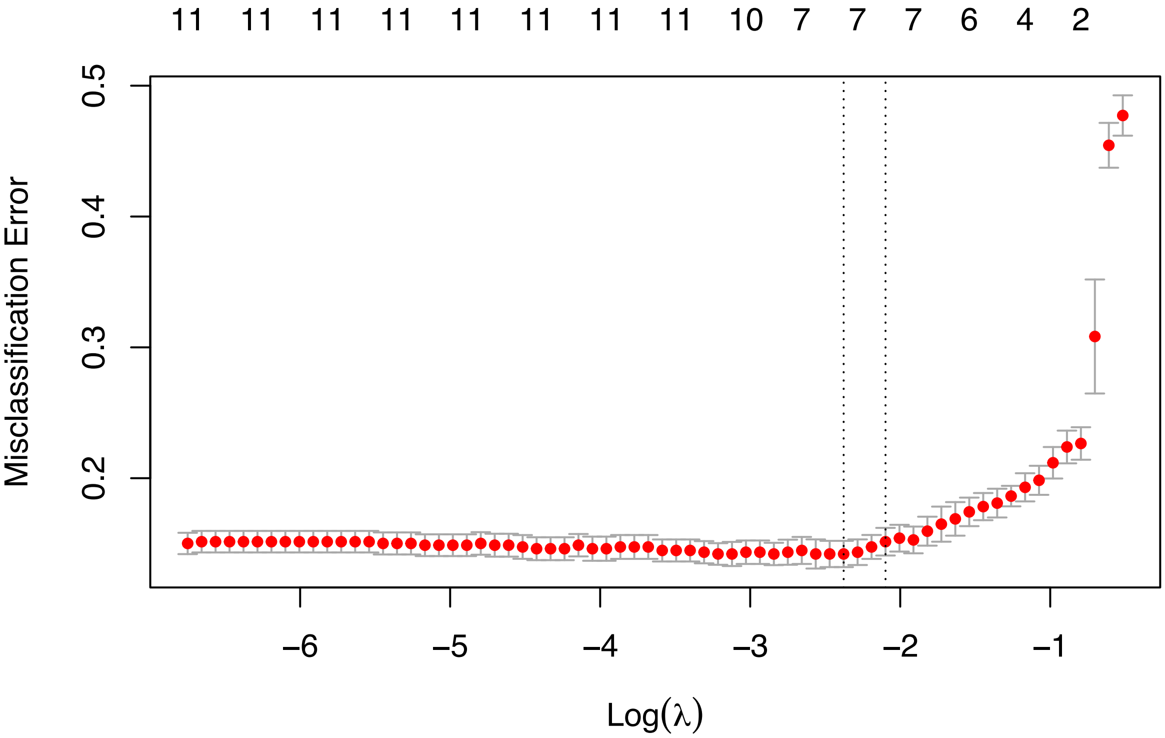
```
## [1] "Age"            "Sex"            "ChestPainType"  "ExerciseAngina"
## [5] "Oldpeak"        "ST_Slope"
```

LASSO

```
x = as.matrix(hd[,1:11])
y = hd$HeartDisease
fit = glmnet(x, y, family = "binomial")
plot(fit, xvar = "dev", label = TRUE)
```



```
cv.out = cv.glmnet(x, y, family = "binomial", type.measure = "class", alpha = 0.5)
plot(cv.out)
```



```
best.lambda <- cv.out$lambda.1se
best.lambda
```

```
## [1] 0.1227147
```

```
co<-coef(cv.out, s = "lambda.1se")
co
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept)     0.555436041
## Age             0.003434928
## Sex             0.344324880
## ChestPainType   0.289812971
## RestingBP       .
## Cholesterol     .
## FastingBS       .
## RestingECG      .
## MaxHR          -0.002576080
## ExerciseAngina  0.738551562
## Oldpeak         0.200582018
## ST_Slope       -0.877223692
```

Fit Model

Predictors selected by AIC

```
model_aic <- glm(as.factor(HeartDisease) ~ Age + Sex + ChestPainType + RestingBP + Cholesterol + Exerci

summary(model_aic)
```

```
##
## Call:
## glm(formula = as.factor(HeartDisease) ~ Age + Sex + ChestPainType +
##     RestingBP + Cholesterol + ExerciseAngina + Oldpeak + ST_Slope,
##     family = "binomial", data = hd)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -4.502549   1.352744  -3.328 0.000873 ***
## Age             0.035113   0.012663   2.773 0.005558 **
## Sex             1.761051   0.288166   6.111 9.89e-10 ***
## ChestPainType   0.740223   0.123948   5.972 2.34e-09 ***
## RestingBP       0.011716   0.006805   1.722 0.085106 .
## Cholesterol     0.003561   0.001890   1.884 0.059622 .
## ExerciseAngina  1.140555   0.246472   4.628 3.70e-06 ***
## Oldpeak         0.363763   0.135902   2.677 0.007436 **
## ST_Slope       -1.821718   0.232626  -7.831 4.84e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1032.6  on 745  degrees of freedom
## Residual deviance:  532.6  on 737  degrees of freedom
## AIC: 550.6
##
## Number of Fisher Scoring iterations: 5
```

```
vif(model_aic)
```

```
##            Age            Sex  ChestPainType      RestingBP    Cholesterol 
##       1.079363       1.102220       1.085802       1.078356       1.048049 
## ExerciseAngina        Oldpeak       ST_Slope 
##       1.145799       1.276588       1.264206
```

Predictors selected by BIC

```
model_bic <- glm(as.factor(HeartDisease) ~ Age + Sex + ChestPainType + ExerciseAngina + Oldpeak + ST_Slo

summary(model_bic)
```

```
## 
## Call:
## glm(formula = as.factor(HeartDisease) ~ Age + Sex + ChestPainType + 
##     ExerciseAngina + Oldpeak + ST_Slope, family = "binomial", 
##     data = hd)
## 
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)    
## (Intercept)    -2.17609    1.00320  -2.169  0.03007 *  
## Age             0.03875    0.01239   3.127  0.00177 ** 
## Sex             1.64061    0.27691   5.925 3.13e-09 ***
## ChestPainType   0.72881    0.12282   5.934 2.96e-09 ***
## ExerciseAngina  1.17905    0.24446   4.823 1.41e-06 ***
## Oldpeak         0.37809    0.13438   2.814  0.00490 ** 
## ST_Slope       -1.81497    0.23141  -7.843 4.40e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 1032.63  on 745  degrees of freedom
## Residual deviance:  539.87  on 739  degrees of freedom
## AIC: 553.87
## 
## Number of Fisher Scoring iterations: 5
```

```
vif(model_bic)
```

```
##            Age            Sex  ChestPainType ExerciseAngina        Oldpeak 
##       1.031274       1.056649       1.069207       1.140508       1.280950 
##       ST_Slope 
##       1.261224
```

Predictors selected by LASSO

```
model_lasso <- glm(as.factor(HeartDisease) ~ Age + MaxHR + ExerciseAngina + Oldpeak + ST_Slope, data=hd

summary(model_lasso)
```

```
## 
## Call:
## glm(formula = as.factor(HeartDisease) ~ Age + MaxHR + ExerciseAngina + 
##     Oldpeak + ST_Slope, family = "binomial", data = hd)
## 
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)    
## (Intercept)     3.325480   1.129903   2.943  0.00325 ** 
## Age             0.025952   0.011830   2.194  0.02826 *  
## MaxHR          -0.011335   0.004688  -2.418  0.01561 *  
## ExerciseAngina  1.541495   0.227917   6.763 1.35e-11 ***
## Oldpeak         0.380263   0.124704   3.049  0.00229 ** 
## ST_Slope       -1.708868   0.213711  -7.996 1.28e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 1032.63  on 745  degrees of freedom
## Residual deviance:  612.45  on 740  degrees of freedom
## AIC: 624.45
## 
## Number of Fisher Scoring iterations: 5
```

```
vif(model_lasso)
```

```
##            Age          MaxHR ExerciseAngina        Oldpeak       ST_Slope 
##       1.104076       1.156175       1.121129       1.264864       1.208059 
```

Predictors selected by considering the results of stepwise selection base on AIC and BIC, and LASSO

```
model_fitted <- glm(as.factor(HeartDisease) ~ Sex + ChestPainType + ExerciseAngina + ST_Slope, data=hd,

summary(model_fitted)
```

```
## 
## Call:
## glm(formula = as.factor(HeartDisease) ~ Sex + ChestPainType + 
##     ExerciseAngina + ST_Slope, family = "binomial", data = hd)
## 
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)    
## (Intercept)      1.0469     0.6328   1.654    0.098 .  
## Sex              1.5998     0.2731   5.859 4.67e-09 ***
## ChestPainType    0.7271     0.1200   6.059 1.37e-09 ***
## ExerciseAngina   1.4100     0.2341   6.023 1.71e-09 ***
## ST_Slope        -2.1831     0.2088 -10.456  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 1032.63  on 745  degrees of freedom
## Residual deviance:  560.82  on 741  degrees of freedom
## AIC: 570.82
## 
## Number of Fisher Scoring iterations: 5
```

```
vif(model_fitted)
```

```
##            Sex  ChestPainType ExerciseAngina       ST_Slope
##       1.054493       1.066563       1.084986       1.079427
```

Use the model selected by stepwise selection base on AIC.

Check Model

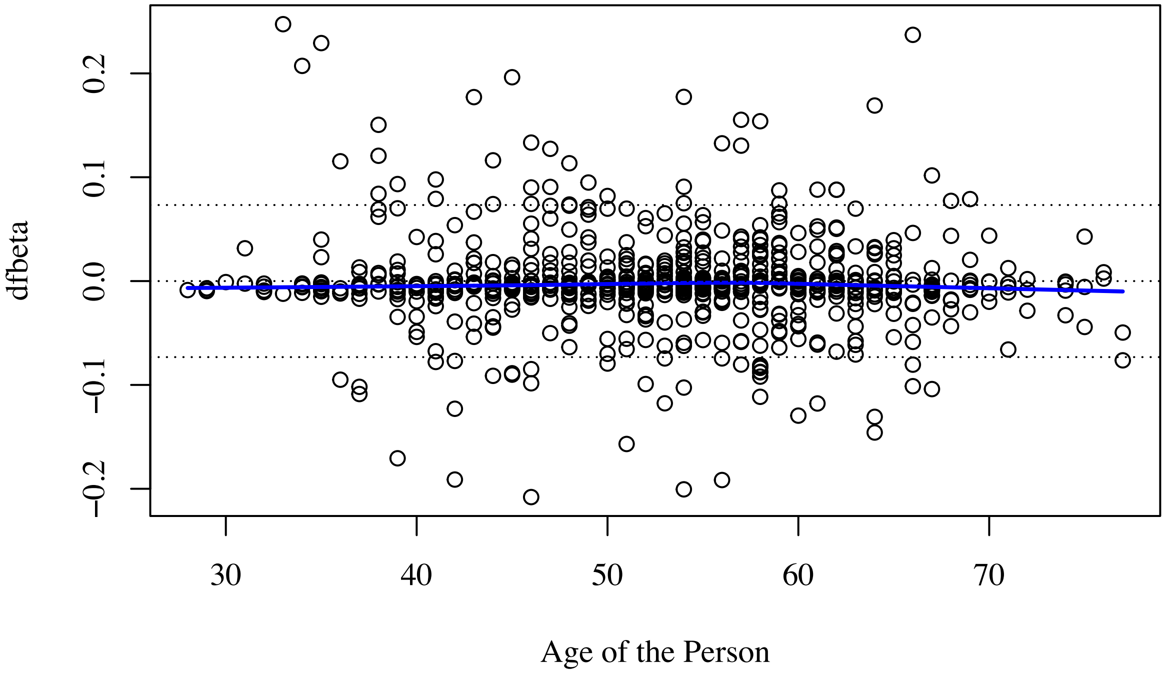
Dfbetas

beta 1 (Age)

```
log.mod.final <- glm(as.factor(HeartDisease) ~ Age + Sex + ChestPainType + RestingBP + Cholesterol + Exe
df.final <- dfbetas(log.mod.final)
head(df.final)
```

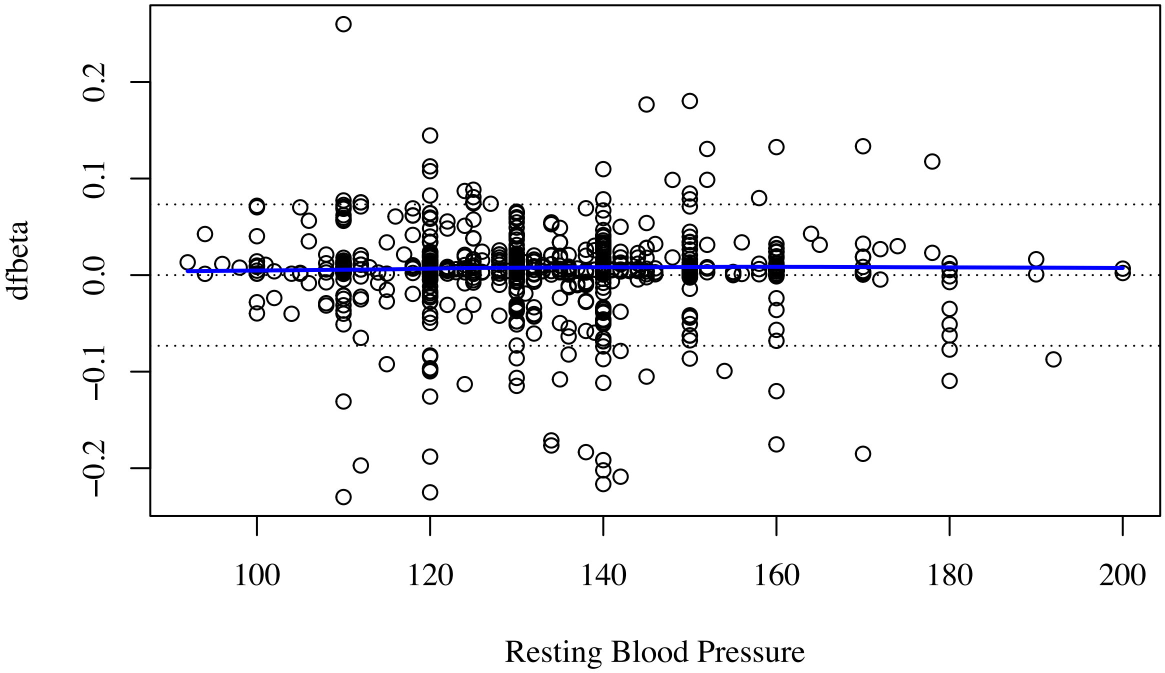
```
##    (Intercept)           Age          Sex ChestPainType     RestingBP
## 1 -0.002857832  0.0171981525 -0.002539332  0.0157114777 -0.0067726207
## 2  0.063945986 -0.1070738891 -0.205756397 -0.0004857738  0.1749808925
## 3 -0.007120961  0.0158373727 -0.001219879  0.0132365623 -0.0002766835
## 4  0.049568035 -0.0620827228 -0.120370516  0.0262495743  0.0263287713
## 5  0.016807374  0.0005324482 -0.004915545  0.0015169717 -0.0317449961
## 6 -0.005610458  0.0261876468 -0.008219629  0.0074222608  0.0124877211
##    Cholesterol ExerciseAngina      Oldpeak     ST_Slope
## 1 -0.008930257   0.0011036215 0.0035933286 -0.009150672
## 2 -0.160183848  -0.0908916552 0.0002000045 -0.056906205
## 3 -0.006420281   0.0004856779 0.0023378801 -0.007680645
## 4 -0.060365736   0.0585515238 0.0189662311 -0.007714901
## 5  0.024320634   0.0132671915 0.0122860493 -0.019556347
## 6 -0.035676393   0.0064260899 0.0039078621 -0.016719074
```

```
par(family = 'serif')
plot(hd$Age, df.final[,1], xlab='Age of the Person',
     ylab='dfbeta')
lines(lowess(hd$Age, df.final[,1]), lwd=2, col='blue')
abline(h=0, lty='dotted')
abline(h=-2/sqrt(nrow(df.final)), lty='dotted')
abline(h=2/sqrt(nrow(df.final)), lty='dotted')
```



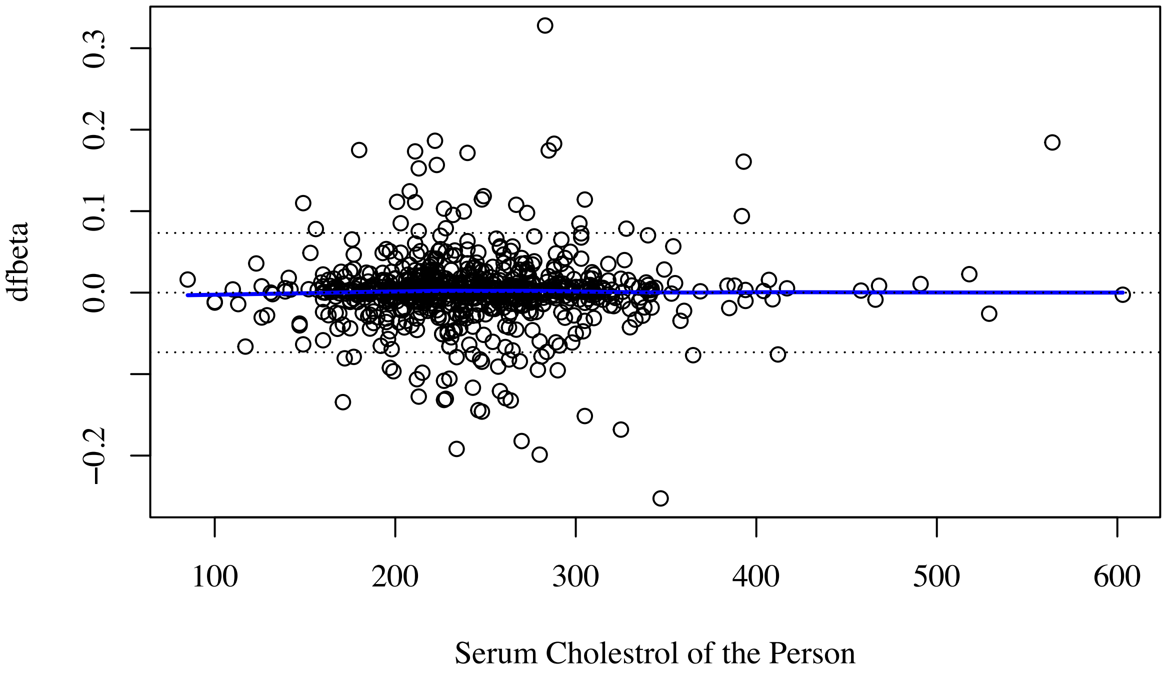
beta 4 (RestingBP)

```
par(family = 'serif')
plot(hd$RestingBP, df.final[,4], xlab='Resting Blood Pressure',
     ylab='dfbeta')
lines(lowess(hd$RestingBP, df.final[,4]), lwd=2, col='blue')
abline(h=0, lty='dotted')
abline(h=-2/sqrt(nrow(df.final)), lty='dotted')
abline(h=2/sqrt(nrow(df.final)), lty='dotted')
```



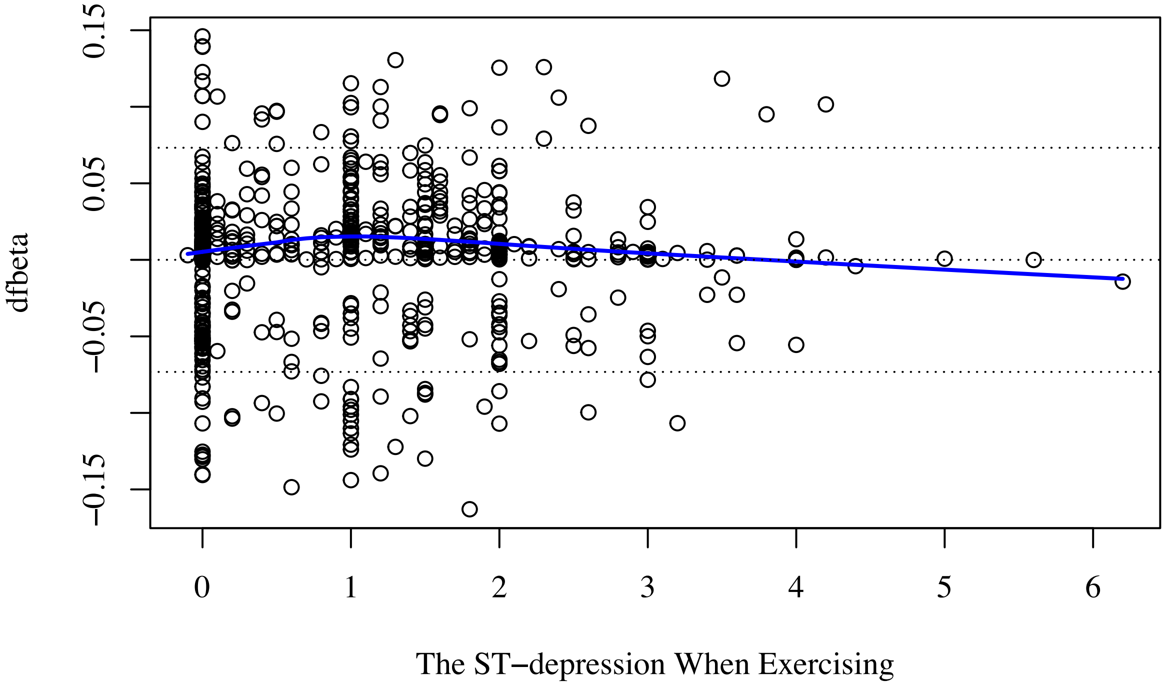
beta 5 (Cholesterol)

```
par(family = 'serif')
plot(hd$Cholesterol, df.final[,5], xlab='Serum Cholestrol of the Person',
     ylab='dfbeta')
lines(lowess(hd$Cholesterol, df.final[,5]), lwd=2, col='blue')
abline(h=0, lty='dotted')
abline(h=-2/sqrt(nrow(df.final)), lty='dotted')
abline(h=2/sqrt(nrow(df.final)), lty='dotted')
```



beta 7 (Oldpeak)

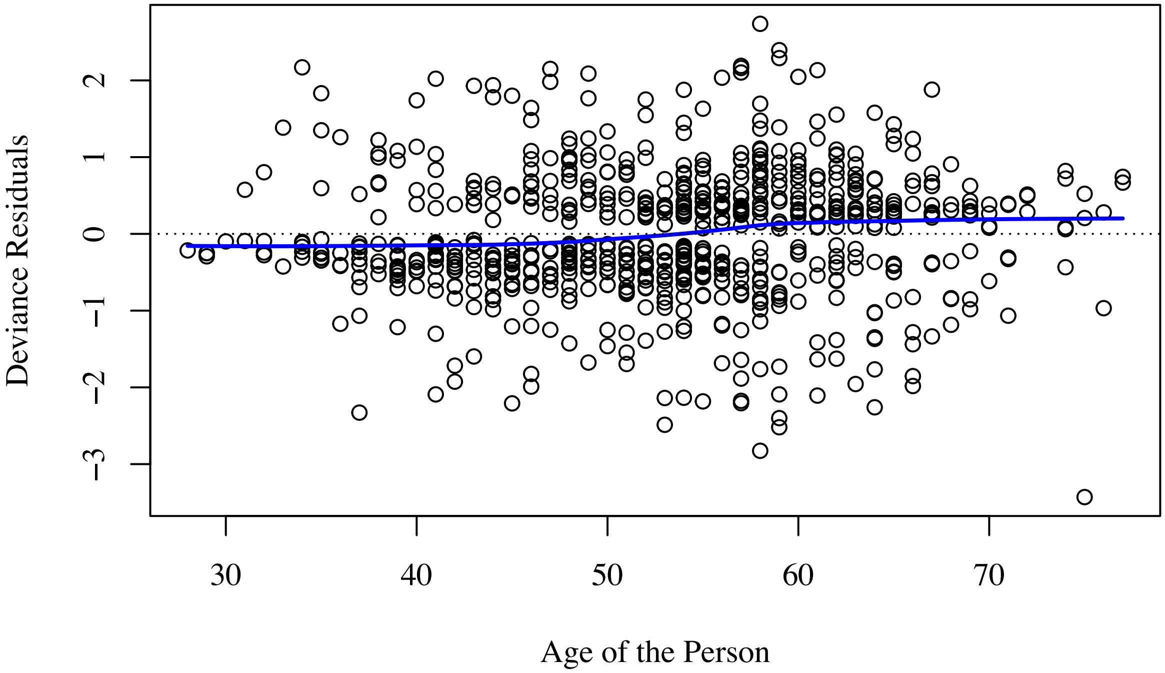
```
par(family = 'serif')
plot(hd$Oldpeak, df.final[,7], xlab='The ST-depression When Exercising',
     ylab='dfbeta')
lines(lowess(hd$Oldpeak, df.final[,7]), lwd=2, col='blue')
abline(h=0, lty='dotted')
abline(h=-2/sqrt(nrow(df.final)), lty='dotted')
abline(h=2/sqrt(nrow(df.final)), lty='dotted')
```



Deviance residuals

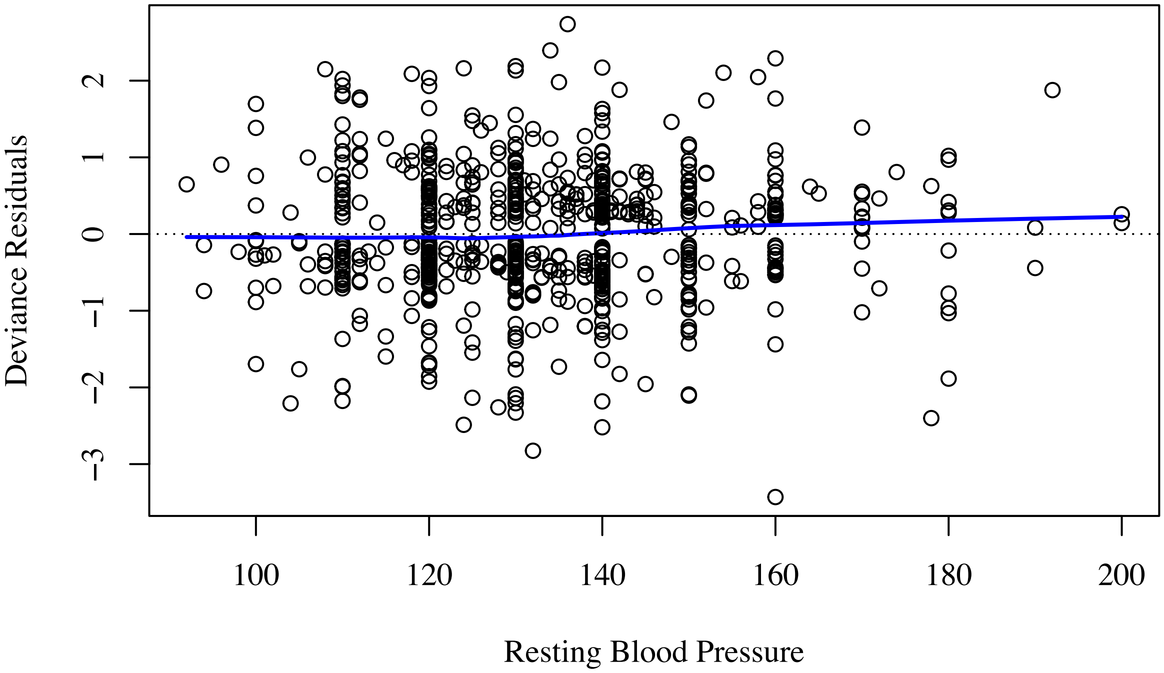
Age

```
res.dev <- residuals(log.mod.final, type = "deviance")
par(family = 'serif')
plot(hd$Age, res.dev, xlab='Age of the Person',
     ylab='Deviance Residuals')
lines(lowess(hd$Age, res.dev), lwd=2, col='blue')
abline(h=0, lty='dotted')
```



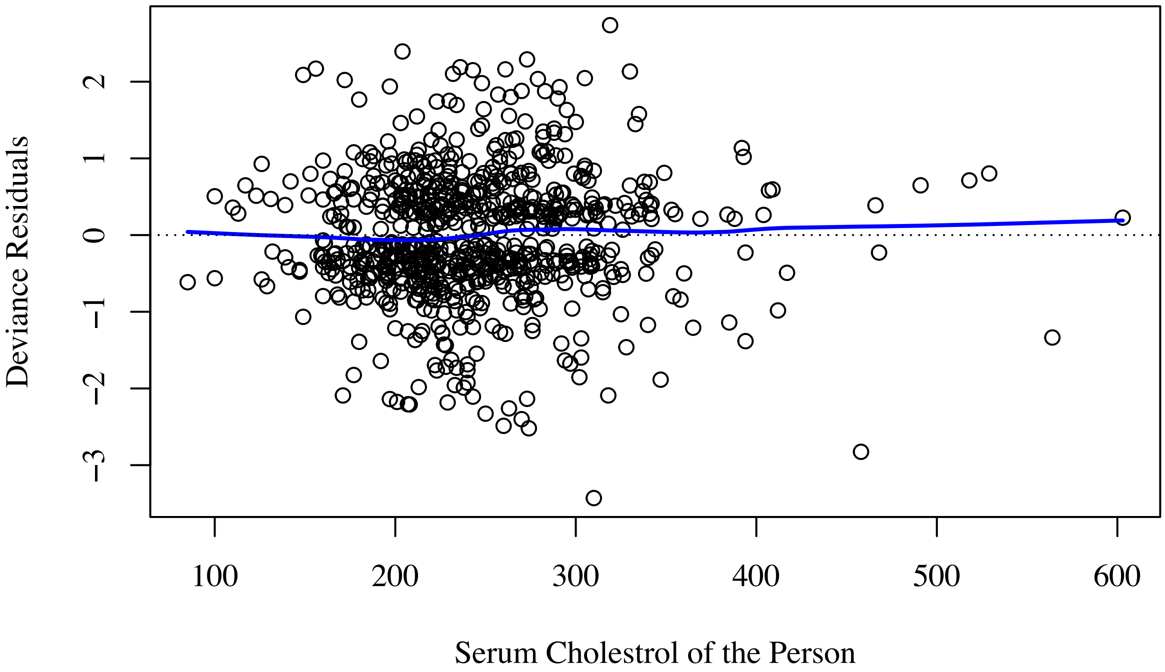
RestingBP

```
res.dev <- residuals(log.mod.final, type = "deviance")
par(family = 'serif')
plot(hd$RestingBP, res.dev, xlab='Resting Blood Pressure',
     ylab='Deviance Residuals')
lines(lowess(hd$RestingBP, res.dev), lwd=2, col='blue')
abline(h=0, lty='dotted')
```



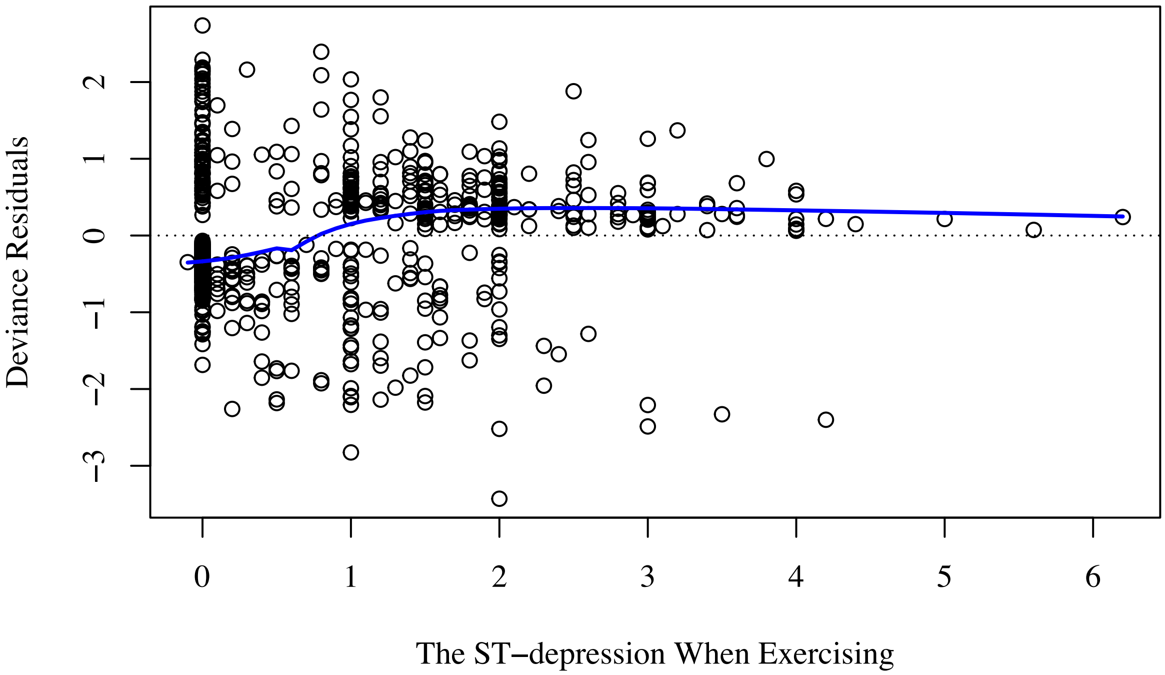
Cholesterol

```
res.dev <- residuals(log.mod.final, type = "deviance")
par(family = 'serif')
plot(hd$Cholesterol, res.dev, xlab='Serum Cholestrol of the Person',
     ylab='Deviance Residuals')
lines(lowess(hd$Cholesterol, res.dev), lwd=2, col='blue')
abline(h=0, lty='dotted')
```



Oldpeak

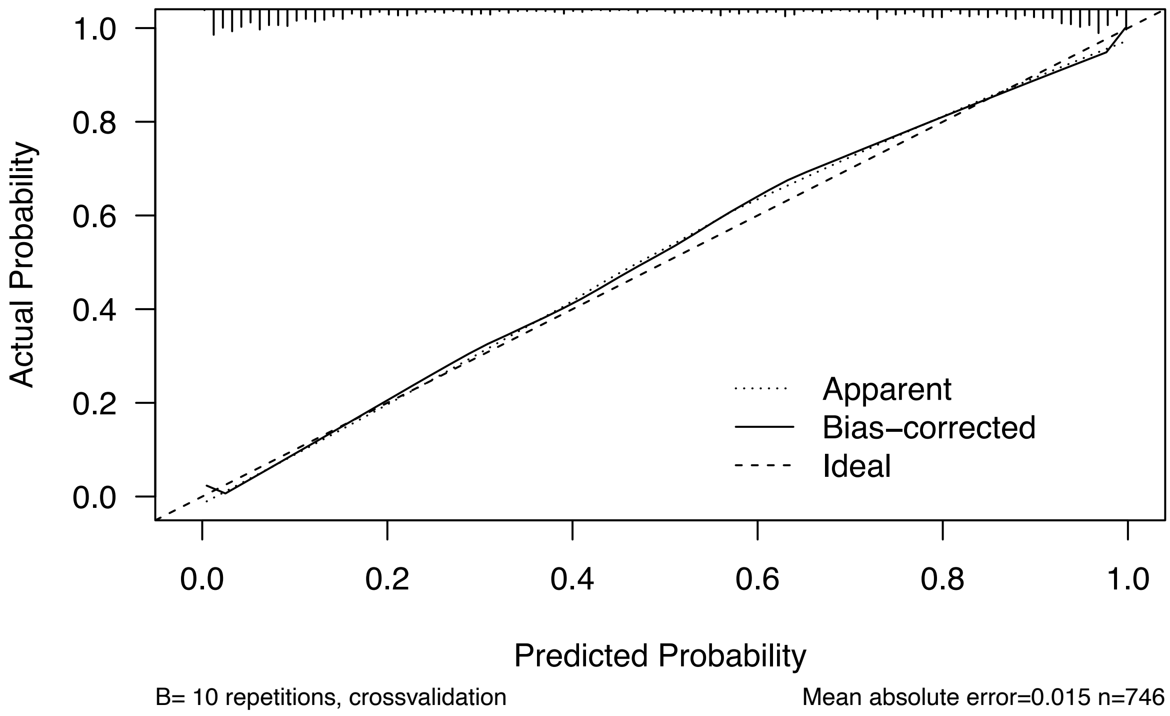
```r
res.dev <- residuals(log.mod.final, type = "deviance")
par(family = 'serif')
plot(hd$Oldpeak, res.dev, xlab='The ST-depression When Exercising',
     ylab='Deviance Residuals')
lines(lowess(hd$Oldpeak, res.dev), lwd=2, col='blue')
abline(h=0, lty='dotted')
```



Vlidation

Calibration Plot

```
hd$HeartDisease <- as.factor(hd$HeartDisease)

## Fit the model with lrm from rms package ##
lrm.final <- lrm(HeartDisease ~ ., data = hd[,which(colnames(hd) %in% c(select_var_aic, "HeartDisease")
cross.calib <- calibrate(lrm.final, method="crossvalidation", B=10) # model calibration
plot(cross.calib, las=1, xlab = "Predicted Probability")
```



```
##
## n=746   Mean absolute error=0.015   Mean squared error=0.00033
## 0.9 Quantile of absolute error=0.027
```

ROC Curve

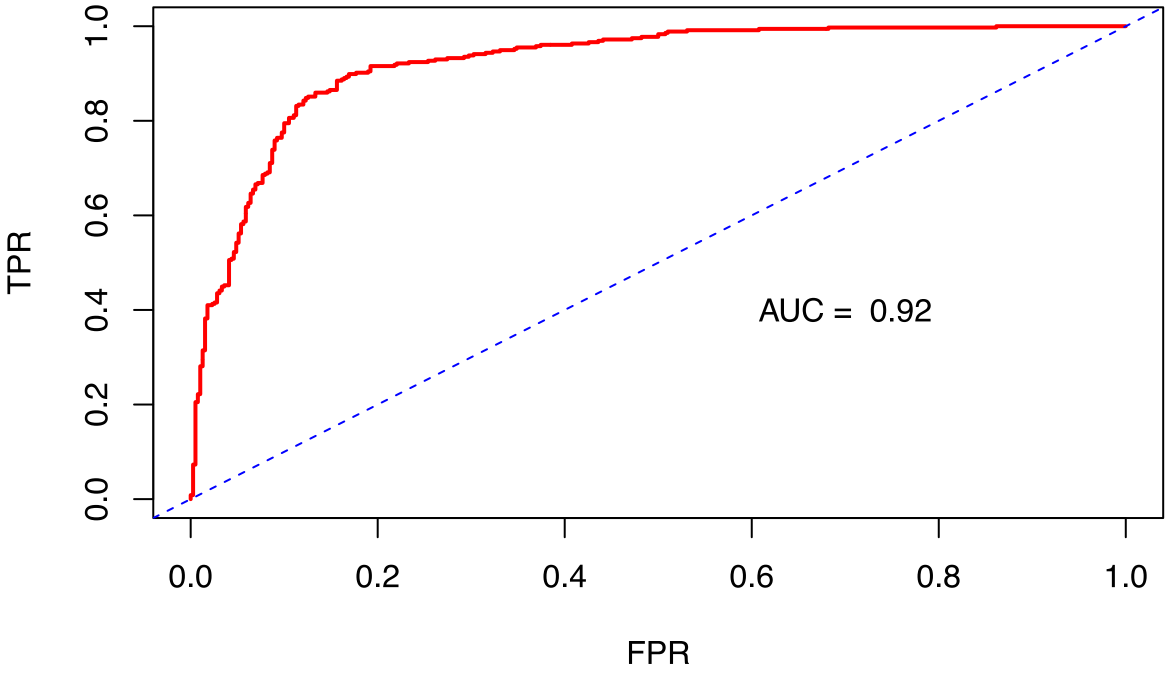
```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
p <- predict(lrm.final, type = "fitted")

roc_logit <- roc(hd$HeartDisease ~ p)
```

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

```
## The True Positive Rate ##
TPR <- roc_logit$sensitivities
## The False Positive Rate ##
FPR <- 1 - roc_logit$specificities

plot(FPR, TPR, xlim = c(0,1), ylim = c(0,1), type = 'l', lty = 1, lwd = 2,col = 'red')
abline(a = 0, b = 1, lty = 2, col = 'blue')
text(0.7,0.4,label = paste("AUC = ", round(auc(roc_logit),2)))
```



```
auc(roc_logit)
```

```
## Area under the curve: 0.9215
```