Heart disease Analysis project

Sengdao_Oudomsihn

4/15/2022

The dataset for this statistical analysis was retrieved from the UCI Machine Learning Repository. The dataset for this statistical analysis consists of 303 observations and 14 variables. However, column names for the dataset were not included when the data was imported from the UCI Machine Learning Repository website. Below are columns names I give to the columns according to the UCI Machine Learning Repository website

Age = age in year Sex = gender of patients; 1=male, 0=female CP_Type = chest pain type (Typical angina, Atypical angina, Non-anginal pain, Asymptomatic) BloodPres = blood pressure (millimeters of mercury (mmHg)) Cholesterol = serum cholesterol in mg/dl Fasting_BP = fasting blood sugar >120 mg/dl (1=true; 0=false) Rest_ECG = resting electrocardiographic results (0=normal, 1= can range from mild symptoms to severe problems, 2: possible or definite left ventricular hypertrophy MaxHR = maximum heart rate achieved Exercise = Exercise induced angina (1= yes, 0=no) ST_Dep = ST Depression induced by exercise relative to rest Slope_ST =The slope of the peak exercise ST segment (1:upsloping, 2: flat sloping and 3: downsloping Num_Vsel = number of major vessels (0-3) Thallium_Stress = Thallium stress result (3: normal, 6: fixed defect, 7: reversible defect HD = Heart disease conditions

this statistical analysis aims not to classify the type of heart condition, but on the other hand, the aim of the paper is to investigate whether; 1. Age has no effect on the heart disease condition of patients 2. ST-Segment Depression has no effects the ST-elevation of heart rate 3. Age and sex are not predictors for heart disease 4. ST-segment Depression (ST_Dep) and ST-elevation of heart rate are predictors for different kinds of chest pains 5. There is no correlation between age and cholesterol 6. There is no correlation between blood pressure and cholesterol 7. There is no correlation between age and maximum heart rate

A few types of statistical analysis models will be used to find the answer for hypotheses. These types of statistical analysis models include one-way ANOVA, two-way ANOVA, and linear regression. Each statistical analysis model is unique in its own way, and each is used differently based on the type of data that are being compared. So the hypotheses for this analysis include many types of data. Therefore I will use different statistical analysis models for each hypothesis testing.

tidyverse package

```
# install.packages("tidyverse")
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✓ ggplot2 3.3.5     ✓ purrr   0.3.4
## ✓ tibble  3.1.6     ✓ dplyr   1.0.8
## ✓ tidyr   1.1.4     ✓ stringr 1.4.0
## ✓ readr   2.0.1     ✓ forcats 0.5.1
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

referenece source of dataset: https://archive.ics.uci.edu/ml/datasets/Heart+Disease (https://archive.ics.uci.edu/ml/datasets/Heart+Disease)

```
urlfile="https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/processed.cleveland.data"
my_data <- read.csv(urlfile)
str(my_data)
```

```
## 'data.frame':    302 obs. of  14 variables:
##  $ X63.0 : num  67 67 37 41 56 62 57 63 53 57 ...
##  $ X1.0  : num  1 1 1 0 1 0 0 1 1 1 ...
##  $ X1.0.1: num  4 4 3 2 2 4 4 4 4 4 ...
##  $ X145.0: num  160 120 130 130 120 140 120 130 140 140 ...
##  $ X233.0: num  286 229 250 204 236 268 354 254 203 192 ...
##  $ X1.0.2: num  0 0 0 0 0 0 0 0 1 0 ...
##  $ X2.0  : num  2 2 0 2 0 2 0 2 2 0 ...
##  $ X150.0: num  108 129 187 172 178 160 163 147 155 148 ...
##  $ X0.0  : num  1 1 0 0 0 0 1 0 1 0 ...
##  $ X2.3  : num  1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 0.4 ...
##  $ X3.0  : num  2 2 3 1 1 3 1 2 3 2 ...
##  $ X0.0.1: chr  "3.0" "2.0" "0.0" "0.0" ...
##  $ X6.0  : chr  "3.0" "7.0" "3.0" "3.0" ...
##  $ X0    : int  2 1 0 0 0 3 0 2 1 0 ...
```

Assign names to the column names

```
colnames(my_data) <- c("Age","Sex","CP_Type", "BloodPres", "Cholesterol", "Fasting_BP",  "Rest_ECG", "MaxH
R", "Exercise",  "ST_Dep", "Slope_ST", "Num_Vsels", "Thallium_Stress", "HD")
str(my_data)
```

```
## 'data.frame':    302 obs. of  14 variables:
##  $ Age            : num  67 67 37 41 56 62 57 63 53 57 ...
##  $ Sex            : num  1 1 1 0 1 0 0 1 1 1 ...
##  $ CP_Type        : num  4 4 3 2 2 4 4 4 4 4 ...
##  $ BloodPres      : num  160 120 130 130 120 140 120 130 140 140 ...
##  $ Cholesterol    : num  286 229 250 204 236 268 354 254 203 192 ...
##  $ Fasting_BP     : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ Rest_ECG       : num  2 2 0 2 0 2 0 2 2 0 ...
##  $ MaxHR          : num  108 129 187 172 178 160 163 147 155 148 ...
##  $ Exercise       : num  1 1 0 0 0 0 1 0 1 0 ...
##  $ ST_Dep         : num  1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 0.4 ...
##  $ Slope_ST       : num  2 2 3 1 1 3 1 2 3 2 ...
##  $ Num_Vsels      : chr  "3.0" "2.0" "0.0" "0.0" ...
##  $ Thallium_Stress: chr  "3.0" "7.0" "3.0" "3.0" ...
##  $ HD             : int  2 1 0 0 0 3 0 2 1 0 ...
```

Checking if there are any missing values in the dataset

```
## Data cleaning
any(is.na(my_data)) # check if there is any missing value in the dataset. TRUE mean yes there are missing
 value in the data frame
```

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

```
my_data[my_data == "?"] <- NA # Replace "?" in any column with NA
colSums(is.na(my_data))# which column has missing value
```

```
##             Age             Sex         CP_Type       BloodPres     Cholesterol
##               0               0               0               0               0
##      Fasting_BP        Rest_ECG           MaxHR        Exercise          ST_Dep
##               0               0               0               0               0
##        Slope_ST       Num_Vsels Thallium_Stress              HD
##               0               4               2               0
```

delete anny missing value that could potentially cause problem for analysis

```
my_data <- na.omit(my_data) # omit the missing value in the dataset and save it back to the dateset
any(is.na(my_data)) # False means no missing value in the dataset
```

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

```
my_data[my_data$CP_Type == 1,]$CP_Type <- "Typical angina"
my_data[my_data$CP_Type == 2,]$CP_Type <- "Atypical angina"
my_data[my_data$CP_Type == 3,]$CP_Type <- "Non-anginal pain"
my_data[my_data$CP_Type == 4,]$CP_Type <- "Asymptomatic"
my_data[my_data$Exercise  == 0,]$Exercise  <- "No_Exercise"
my_data[my_data$Exercise  == 1,]$Exercise  <- "Yes_Exercise"
my_data[my_data$Slope_ST == 1,]$Slope_ST <- "Upsloping"
my_data[my_data$Slope_ST == 2,]$Slope_ST <- "Flatsloping"
my_data[my_data$Slope_ST == 3,]$Slope_ST <- "Downsloping"
my_data[my_data$Sex == 1,]$Sex <- "Male"
my_data[my_data$Sex == 0,]$Sex <- "Female"
my_data[my_data$HD == 0,]$HD <- "Healthy"
my_data[my_data$HD == 1,]$HD <- "Unhealthy"
my_data[my_data$HD == 2,]$HD <- "Unhealthy"
my_data[my_data$HD == 3,]$HD <- "Unhealthy"
my_data[my_data$HD == 4,]$HD <- "Unhealthy"
str(my_data)
```

```
## 'data.frame':    296 obs. of  14 variables:
##  $ Age            : num  67 67 37 41 56 62 57 63 53 57 ...
##  $ Sex            : chr  "Male" "Male" "Male" "Female" ...
##  $ CP_Type        : chr  "Asymptomatic" "Asymptomatic" "Non-anginal pain" "Atypical angina" ...
##  $ BloodPres      : num  160 120 130 130 120 140 120 130 140 140 ...
##  $ Cholesterol    : num  286 229 250 204 236 268 354 254 203 192 ...
##  $ Fasting_BP     : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ Rest_ECG       : num  2 2 0 2 0 2 0 2 2 0 ...
##  $ MaxHR          : num  108 129 187 172 178 160 163 147 155 148 ...
##  $ Exercise       : chr  "Yes_Exercise" "Yes_Exercise" "No_Exercise" "No_Exercise" ...
##  $ ST_Dep         : num  1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 0.4 ...
##  $ Slope_ST       : chr  "Flatsloping" "Flatsloping" "Downsloping" "Upsloping" ...
##  $ Num_Vsels      : chr  "3.0" "2.0" "0.0" "0.0" ...
##  $ Thallium_Stress: chr  "3.0" "7.0" "3.0" "3.0" ...
##  $ HD             : chr  "Unhealthy" "Unhealthy" "Healthy" "Healthy" ...
##  - attr(*, "na.action")= 'omit' Named int [1:6] 87 166 192 266 287 302
##   ..- attr(*, "names")= chr [1:6] "87" "166" "192" "266" ...
```

convert some variables in the dataset into appropiate data type for analysis

```
# Now let's convert some variables into factor
my_data$Age <- as.integer(my_data$Age)
my_data$Sex <- as.factor(my_data$Sex)
my_data$CP_Type <- as.factor(my_data$CP_Type)
my_data$Fasting_BP <- as.factor(my_data$Fasting_BP)
my_data$Rest_ECG <- as.factor(my_data$Rest_ECG)
my_data$Exercise <- as.factor(my_data$Exercise)
my_data$Slope_ST <- as.factor(my_data$Slope_ST)
my_data$Num_Vsels <- as.factor(my_data$Num_Vsels)
my_data$Thallium_Stress <- as.factor(my_data$Thallium_Stress)
my_data$BloodPres <- as.numeric(my_data$BloodPres)
my_data$Cholesterol <- as.numeric(my_data$Cholesterol)
my_data$MaxHR <- as.numeric(my_data$MaxHR)
my_data$HD <- as.factor(my_data$HD)
str(my_data)
```

```
## 'data.frame':    296 obs. of  14 variables:
##  $ Age            : int  67 67 37 41 56 62 57 63 53 57 ...
##  $ Sex            : Factor w/ 2 levels "Female","Male": 2 2 2 1 2 1 1 2 2 2 ...
##  $ CP_Type        : Factor w/ 4 levels "Asymptomatic",..: 1 1 3 2 2 1 1 1 1 1 ...
##  $ BloodPres      : num  160 120 130 130 120 140 120 130 140 140 ...
##  $ Cholesterol    : num  286 229 250 204 236 268 354 254 203 192 ...
##  $ Fasting_BP     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 1 ...
##  $ Rest_ECG       : Factor w/ 3 levels "0","1","2": 3 3 1 3 1 3 1 3 3 1 ...
##  $ MaxHR          : num  108 129 187 172 178 160 163 147 155 148 ...
##  $ Exercise       : Factor w/ 2 levels "No_Exercise",..: 2 2 1 1 1 1 2 1 2 1 ...
##  $ ST_Dep         : num  1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 0.4 ...
##  $ Slope_ST       : Factor w/ 3 levels "Downsloping",..: 2 2 1 3 3 1 3 2 1 2 ...
##  $ Num_Vsels      : Factor w/ 4 levels "0.0","1.0","2.0",..: 4 3 1 1 1 3 1 2 1 1 ...
##  $ Thallium_Stress: Factor w/ 3 levels "3.0","6.0","7.0": 1 3 1 1 1 1 1 3 3 2 ...
##  $ HD             : Factor w/ 2 levels "Healthy","Unhealthy": 2 2 1 1 1 2 1 2 2 1 ...
##  - attr(*, "na.action")= 'omit' Named int [1:6] 87 166 192 266 287 302
##   ..- attr(*, "names")= chr [1:6] "87" "166" "192" "266" ...
```

Exploratory Data Analysis

Histogram to see distribution of data in the dataset

```
set.seed(123)
par(mfrow=c(2,3))
hist(my_data$MaxHR, breaks = 20, col="skyblue", main="Heart Rate Maximum", xlab = "Heart Rate")
hist(my_data$BloodPres,breaks = 20, col="skyblue", main="Blood Pressure", xlab = "Blood Pressure")
hist(my_data$Cholesterol, breaks = 20,col="skyblue", main = "Cholesterol Level", xlab = "Cholesterol")
hist(my_data$Age, breaks = 20, col="skyblue", main = "Age in year", xlab = "Age")
hist(my_data$ST_Dep, breaks = 20, col="skyblue", main = "ST Depression", xlab = "ST Depressoin")
```

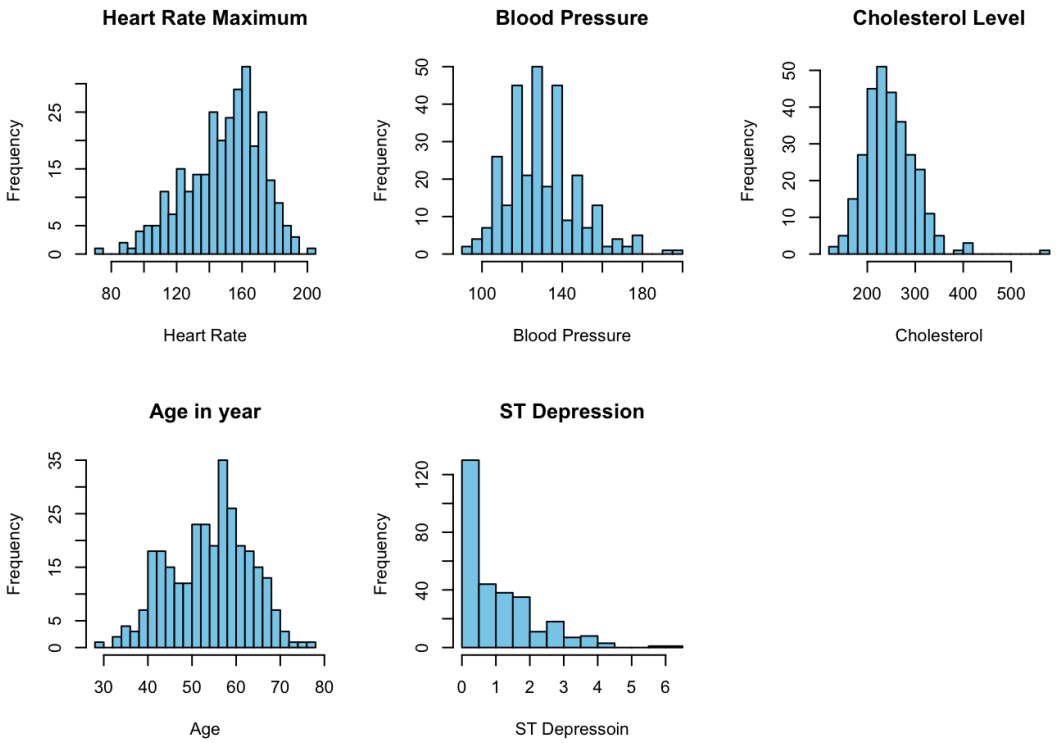
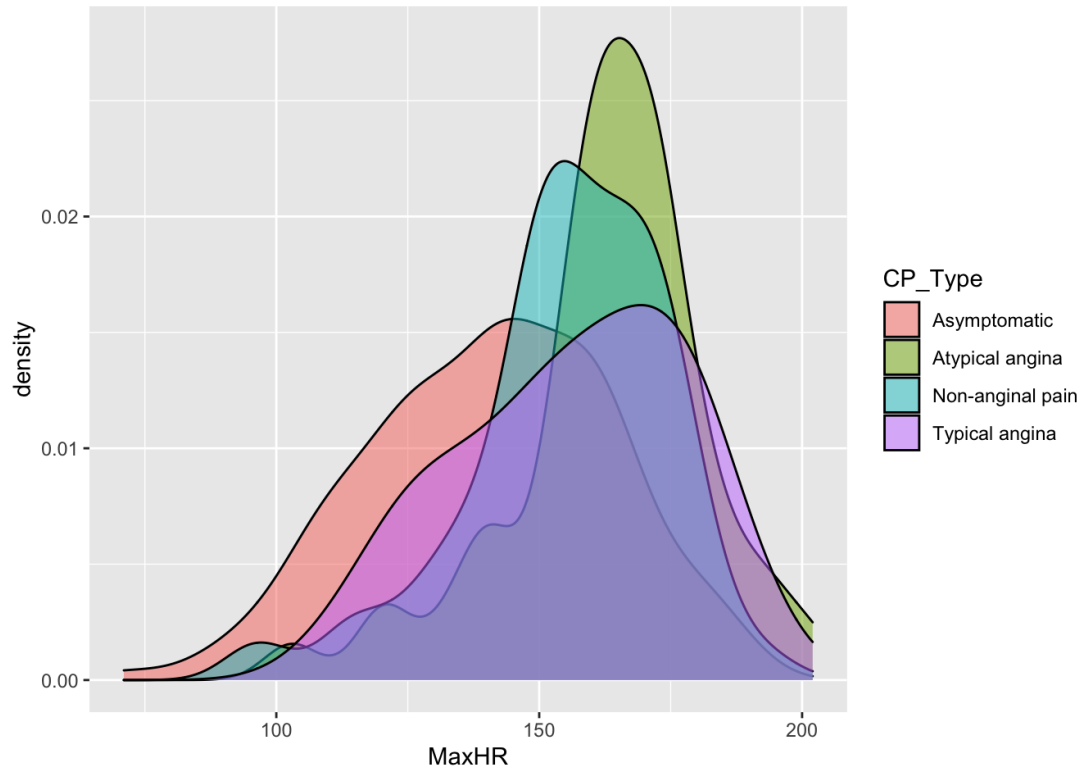


Figure1: Histogram of distribution of data for maximum heart rate, blood pressure, cholesterol level, age, and ST depression for patients in the dataset.

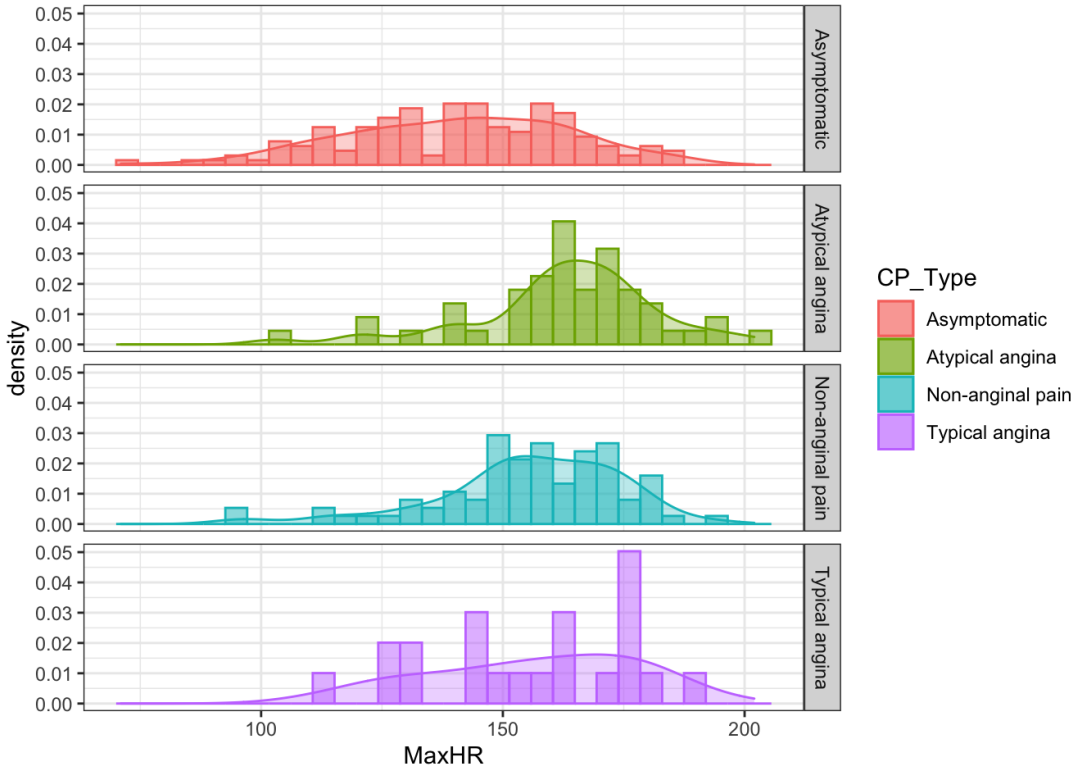
density of data distribution in chest pain

```
ggplot(my_data,aes(x=MaxHR, fill=CP_Type))+geom_density(alpha=.5)
```

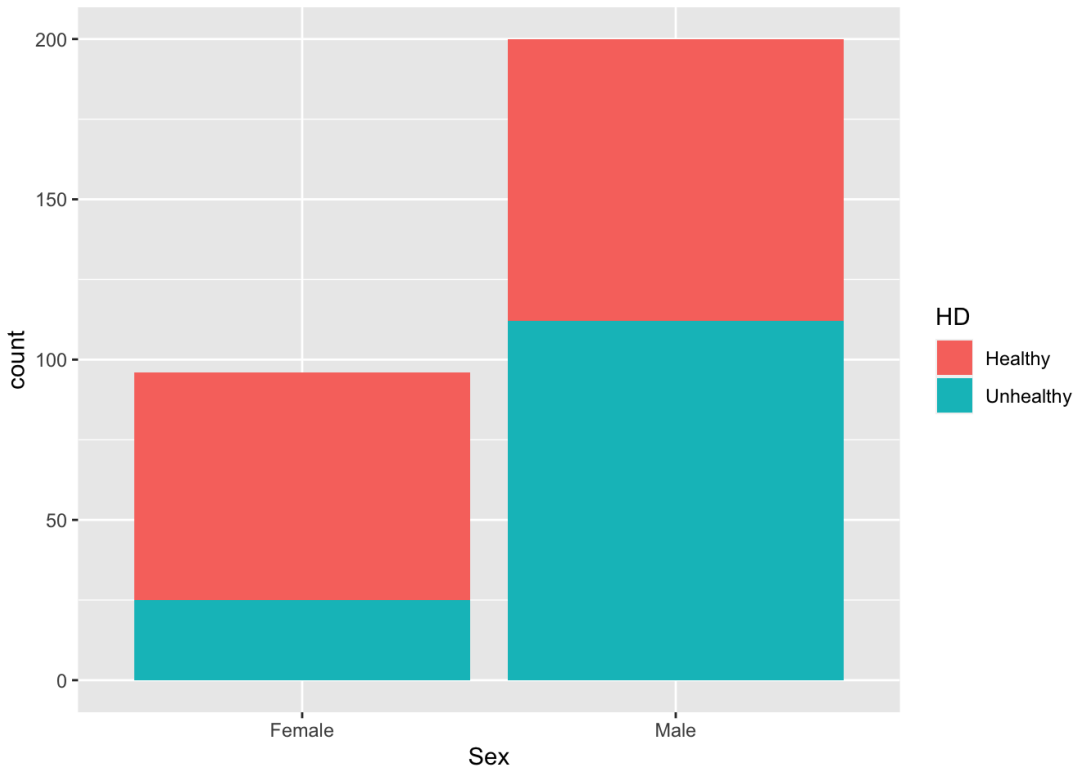



```
ggplot(data = my_data, aes(x=MaxHR, color=CP_Type, fill= CP_Type))+
  geom_histogram(aes(y=..density..), alpha=0.5,
                 position = "identity")+
  geom_density(alpha=.3)+
  facet_grid(CP_Type~.)+
  theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(data=my_data)+geom_bar(mapping = aes(x=Sex, fill=HD))
```




```
my_data %>% group_by(CP_Type) %>%
  count() %>%
  arrange(desc(n))
```

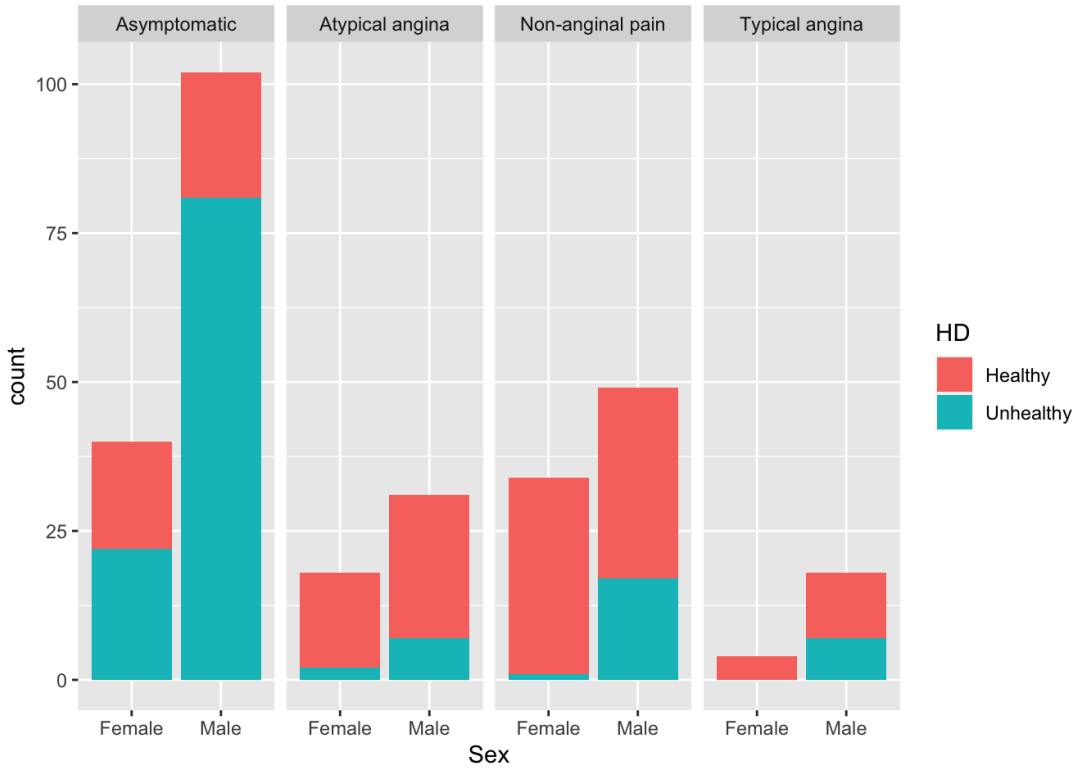
```
## # A tibble: 4 × 2
## # Groups:   CP_Type [4]
##   CP_Type              n
##   <fct>            <int>
## 1 Asymptomatic       142
## 2 Non-anginal pain    83
## 3 Atypical angina     49
## 4 Typical angina      22
```

```
my_data %>%
  filter(CP_Type=="Asymptomatic") %>%
  group_by(HD, Sex, CP_Type) %>%
  summarise(avg_HR=mean(MaxHR),
            median_HR=median(MaxHR),
            avg_age=mean(Age),
            median_age=median(Age)
            )
```

```
## `summarise()` has grouped output by 'HD', 'Sex'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 4 × 7
## # Groups:   HD, Sex [4]
##   HD        Sex    CP_Type      avg_HR median_HR avg_age median_age
##   <fct>     <fct>  <fct>         <dbl>     <dbl>   <dbl>      <dbl>
## 1 Healthy   Female Asymptomatic   148.       153    55.2         57
## 2 Healthy   Male   Asymptomatic   156        160    53.1         53
## 3 Unhealthy Female Asymptomatic   143.       146.   59.1         60.5
## 4 Unhealthy Male   Asymptomatic   134.       132    55.8         57
```

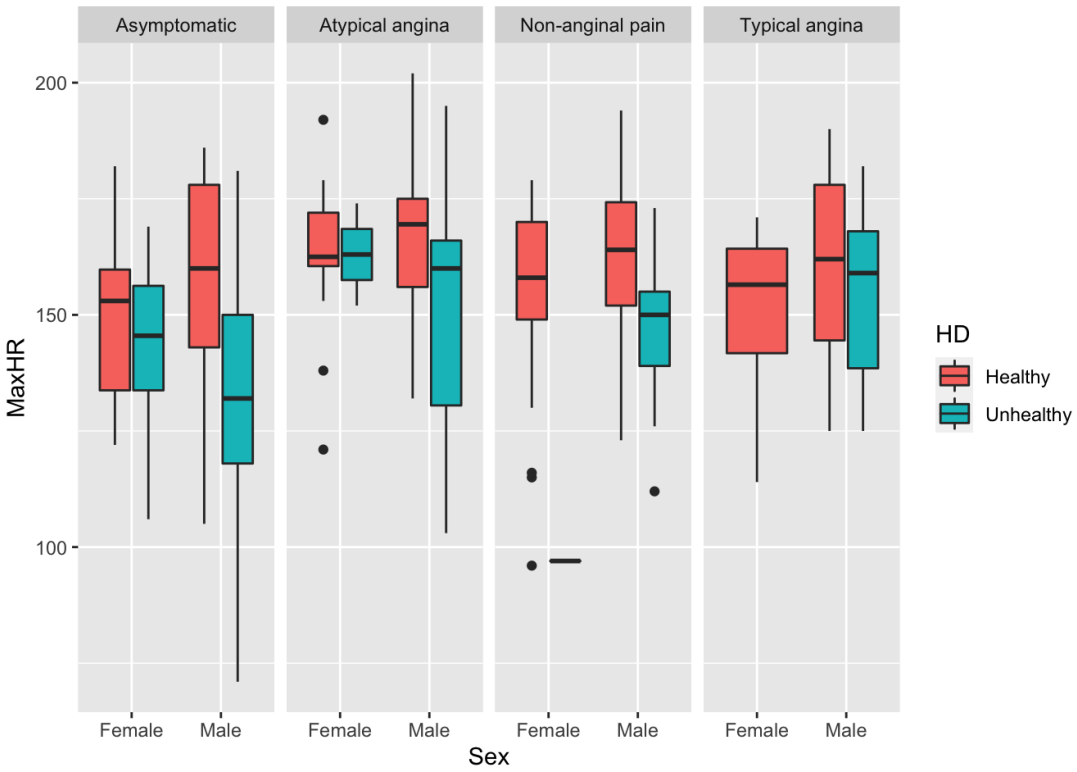
```
ggplot(data=my_data)+geom_bar(mapping = aes(x=Sex, fill=HD))+facet_grid(~CP_Type)
```




```
my_data %>%
  filter(CP_Type=="Asymptomatic") %>%
  group_by(HD, Sex, CP_Type) %>%
  summarise_if(is.numeric, median)
```

```
## # A tibble: 4 × 8
## # Groups:   HD, Sex [4]
##   HD        Sex    CP_Type        Age BloodPres Cholesterol MaxHR ST_Dep
##   <fct>     <fct>  <fct>        <dbl>     <dbl>       <dbl> <dbl>  <dbl>
## 1 Healthy   Female Asymptomatic  57        130          251   153    0.35
## 2 Healthy   Male   Asymptomatic  53        128          228   160    0.2
## 3 Unhealthy Female Asymptomatic  60.5      148.         268.  146.   1.85
## 4 Unhealthy Male   Asymptomatic  57        128          249   132    1.4
```

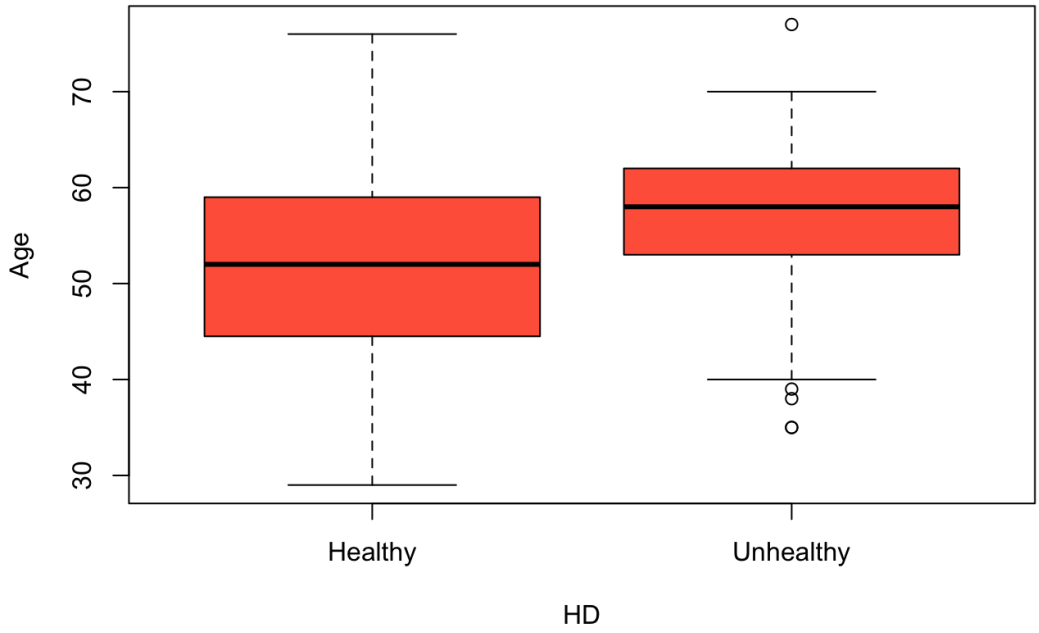
```
ggplot(data = my_data, aes(x= Sex, y= MaxHR, fill= HD))+geom_boxplot()+
  facet_grid(.~CP_Type)
```



Statistical Analysis | Finding insight in data

ONE-Way ANOVA

```
# One-Way ANOVA
boxplot(Age~HD, data=my_data, col='tomato')
```




```
ONE_ANOVA1 <- aov(Age~HD, data=my_data)
summary(ONE_ANOVA1)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## HD            1   1286  1286.1   16.52 6.17e-05 ***
## Residuals   294  22884    77.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

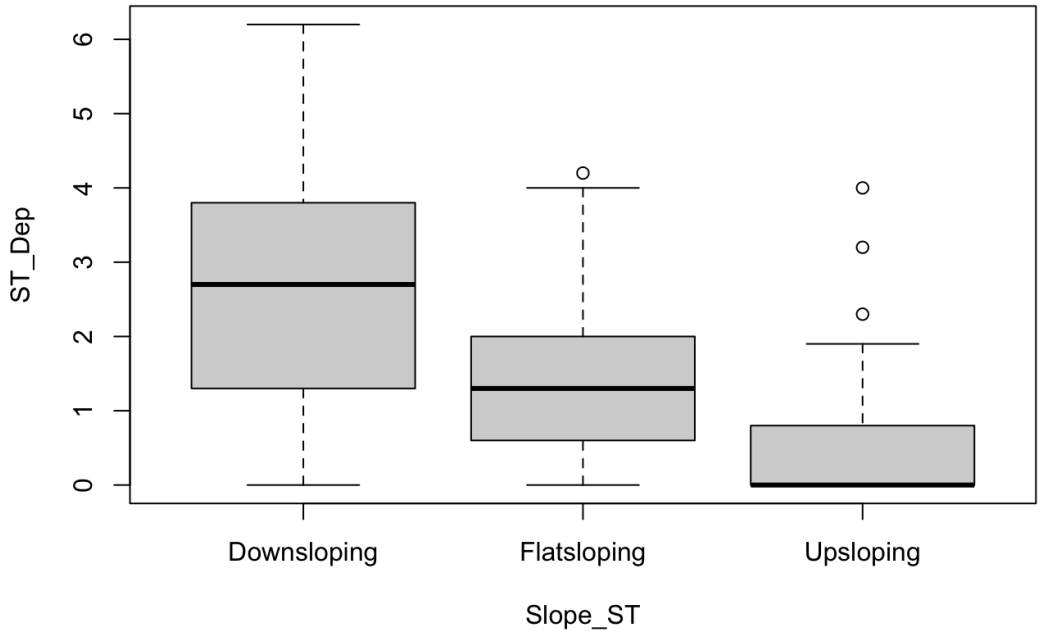
```
TukeyHSD(ONE_ANOVA1)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Age ~ HD, data = my_data)
##
## $HD
##                       diff      lwr      upr    p adj
## Unhealthy-Healthy 4.180508 2.156477 6.204539 6.17e-05
```

Older age patients have higer unhealthy heart condition than younger age patients

Report: “Age of patients are significantly different for heart disease condition(One-way anova, F_1,294=16,52, p<0.0000617)”

```
set.seed(11)
boxplot(ST_Dep~Slope_ST, data=my_data)
```




```
ONE_ANOVA2 <- aov(ST_Dep~Slope_ST, data = my_data)
ONE_ANOVA2
```

```
## Call:
##    aov(formula = ST_Dep ~ Slope_ST, data = my_data)
##
## Terms:
##                 Slope_ST Residuals
## Sum of Squares  134.2331  266.7264
## Deg. of Freedom        2       293
##
## Residual standard error: 0.9541116
## Estimated effects may be unbalanced
```

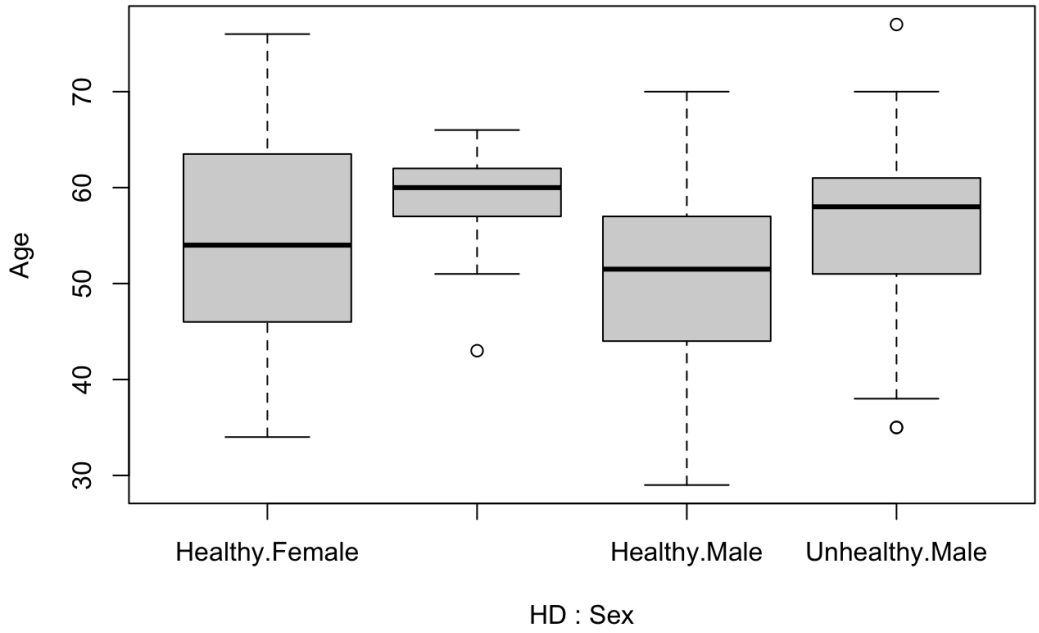
```
summary(ONE_ANOVA2)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Slope_ST      2  134.2   67.12   73.73 <2e-16 ***
## Residuals   293  266.7    0.91
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

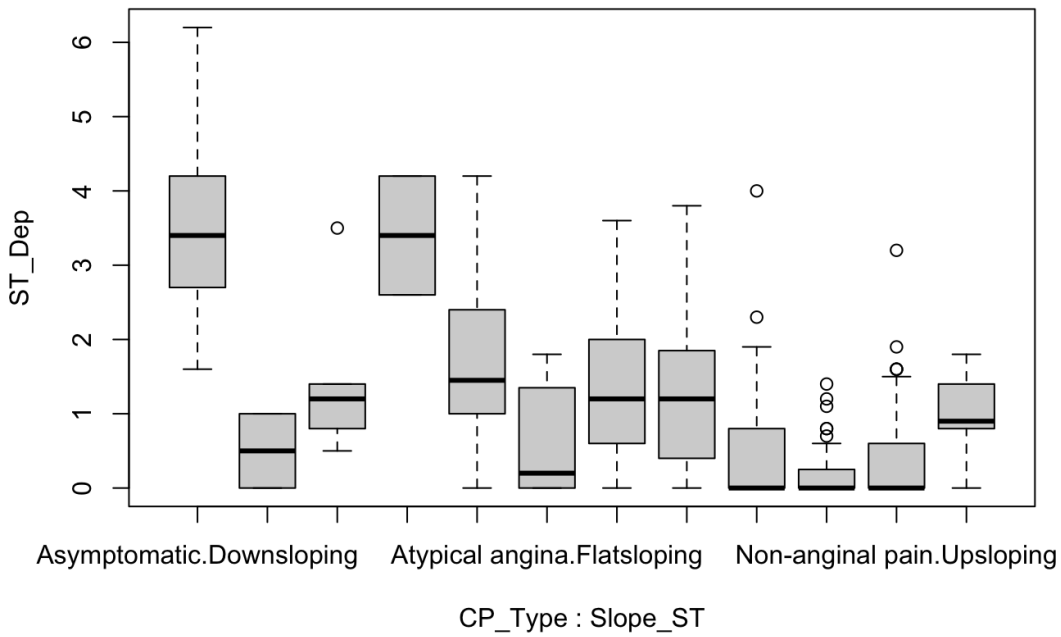
Report: “ST Depression differed significantly among the three heart rate slope(One-way ANOVA, F_2,293 =73.73, P>0.000000000000002 or 2e-16)”

Two-Way ANOVA

```
# two-way anova
boxplot(Age~HD+Sex, data = my_data)
```




```
boxplot(ST_Dep~CP_Type+Slope_ST, data = my_data)
```




```
set.seed(12)
TWO_ANOVA1 <- aov(Age~HD*Sex, data = my_data)
TWO_ANOVA1
```

```
## Call:
##    aov(formula = Age ~ HD * Sex, data = my_data)
##
## Terms:
##                        HD       Sex    HD:Sex Residuals
## Sum of Squares   1286.128   669.240     8.026 22206.553
## Deg. of Freedom         1         1         1       292
##
## Residual standard error: 8.720656
## Estimated effects may be unbalanced
```

```
summary(TWO_ANOVA1)
```

```
##              Df Sum Sq Mean Sq F value  Pr(>F)
## HD            1   1286  1286.1  16.912 5.1e-05 ***
## Sex           1    669   669.2   8.800 0.00326 **
## HD:Sex        1      8     8.0   0.106 0.74552
## Residuals   292  22207    76.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Report: A two-way ANOVA analysis showed that heart condition(HD) was significantly affected by age of patients($F_{1,292}=16.963$, $p< 0.00051$) and sex of patients($F_{1,292}=8.800$, $p<0.00326$), with no significant different($F_{1,292}=0.106$, $p=0.74552$)

```
set.seed(13)
TWO_ANOVA2 <- aov(ST_Dep~CP_Type*Slope_ST, data = my_data)
TWO_ANOVA2
```

```
## Call:
##    aov(formula = ST_Dep ~ CP_Type * Slope_ST, data = my_data)
##
## Terms:
##                    CP_Type  Slope_ST CP_Type:Slope_ST Residuals
## Sum of Squares   48.25395 104.13174         23.68134 224.89243
## Deg. of Freedom         3         2                 6       284
##
## Residual standard error: 0.8898734
## Estimated effects may be unbalanced
```

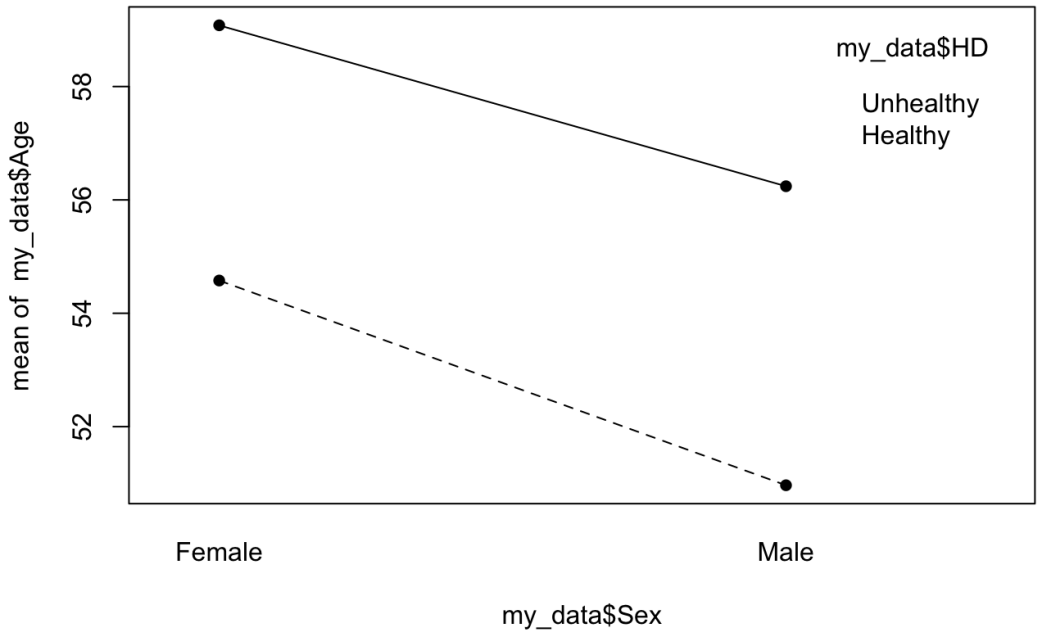
```
summary(TWO_ANOVA2)
```

```
##                   Df Sum Sq Mean Sq F value   Pr(>F)
## CP_Type            3  48.25   16.08  20.312 5.92e-12 ***
## Slope_ST           2 104.13   52.07  65.750  < 2e-16 ***
## CP_Type:Slope_ST   6  23.68    3.95   4.984 7.14e-05 ***
## Residuals        284 224.89    0.79
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

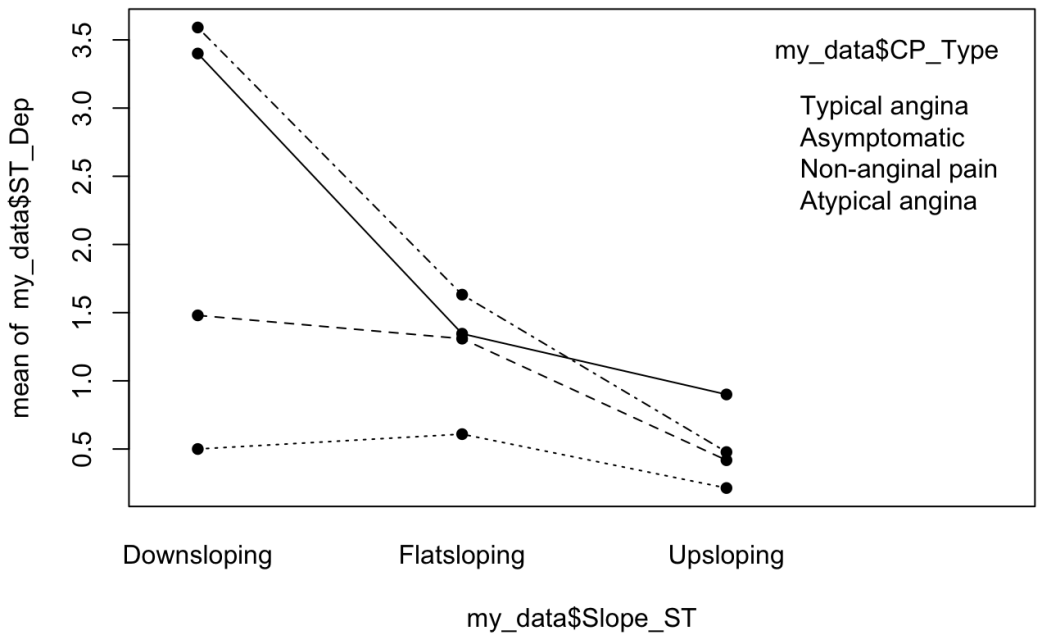
Report: A two-way ANOVA analysis showed that chest pain type(CP_Type) was significantly affected by ST Depression of patients($F_{3,284}=20.312$, $p<0.000000000592$ or 5.92e-12) and

slope of heart rate(Slope_ST) (F_2,284=65.750, p<0.000000000000002 or 2e-16), with a significant different(F_6,284=4.984, p<0.00014)

```
# Chest pain type as prediction for heart disease
interaction.plot(x.factor=my_data$Sex, trace.factor = my_data$HD,my_data$Age, type="o",pch=16)
```




```
# Slope of heart rate as prediction
interaction.plot(x.factor=my_data$Slope_ST,trace.factor = my_data$CP_Type,my_data$ST_Dep, type="o",pch=16)
```



Linear Regression | Finding trends and correlation in data

```
set.seed(14)
linear_model1 <- lm(data = my_data, Cholesterol~Age)
summary(linear_model1)
```

```
##
## Call:
## lm(formula = Cholesterol ~ Age, data = my_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -124.315  -33.136   -5.525   28.093  301.957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 183.4632    18.1525   10.11  < 2e-16 ***
## Age           1.1728     0.3285    3.57 0.000417 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 51.07 on 294 degrees of freedom
## Multiple R-squared:  0.04155,    Adjusted R-squared:  0.03829
## F-statistic: 12.75 on 1 and 294 DF,  p-value: 0.0004165
```

```
cor.test(my_data$Age, my_data$Cholesterol)
```

```
##
##  Pearson's product-moment correlation
##
## data:  my_data$Age and my_data$Cholesterol
## t = 3.5702, df = 294, p-value = 0.0004165
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.09197918 0.31063179
## sample estimates:
##       cor
## 0.2038462
```

Report: There was a significant relationship between age and cholesterol level of patients(linear regression, r^2=0.04155, F_1,294=12.75, p=0.0004165). The correlations between these two variables was positive(r=0.2038462)

```
# Create linear model prediction for the confidence interval band
prediction <- predict.lm(linear_model1, data=my_data, interval = 'confidence',level=0.98)
prediction <- data.frame(prediction)
prediction$Age <- my_data$Age
prediction
```

```
##          fit      lwr      upr Age
## 1   262.0433 250.1996 273.8870  67
## 2   262.0433 250.1996 273.8870  67
## 3   226.8582 211.7148 242.0015  37
## 4   231.5495 219.0579 244.0411  41
## 5   249.1421 242.1051 256.1790  56
## 6   256.1791 247.1620 265.1961  62
## 7   250.3149 243.1132 257.5166  57
## 8   257.3519 247.8262 266.8776  63
## 9   245.6235 238.5832 252.6639  53
## 10  250.3149 243.1132 257.5166  57
## 11  249.1421 242.1051 256.1790  56
## 12  249.1421 242.1051 256.1790  56
## 13  235.0680 224.4153 245.7207  44
## 14  244.4507 237.2435 251.6579  52
## 15  250.3149 243.1132 257.5166  57
## 16  239.7594 231.1999 248.3188  48
## 17  246.7964 239.8416 253.7512  54
## 18  239.7594 231.1999 248.3188  48
## 19  240.9322 232.7981 249.0663  49
## 20  258.5248 248.4574 268.5921  64
## 21  251.4877 244.0452 258.9303  58
## 22  251.4877 244.0452 258.9303  58
## 23  251.4877 244.0452 258.9303  58
## 24  253.8334 245.7101 261.9567  60
## 25  242.1050 234.3434 249.8666  50
## 26  251.4877 244.0452 258.9303  58
## 27  260.8704 249.6402 272.1007  66
## 28  233.8952 222.6486 245.1417  43
## 29  230.3767 217.2394 243.5140  40
## 30  264.3889 251.2693 277.5086  69
## 31  253.8334 245.7101 261.9567  60
## 32  258.5248 248.4574 268.5921  64
## 33  252.6606 244.9082 260.4129  59
## 34  235.0680 224.4153 245.7207  44
## 35  232.7223 220.8618 244.5829  42
## 36  233.8952 222.6486 245.1417  43
## 37  250.3149 243.1132 257.5166  57
## 38  247.9692 241.0155 254.9229  55
## 39  255.0062 246.4589 263.5536  61
## 40  259.6976 249.0607 270.3345  65
## 41  230.3767 217.2394 243.5140  40
## 42  266.7346 252.2881 281.1811  71
## 43  252.6606 244.9082 260.4129  59
## 44  255.0062 246.4589 263.5536  61
## 45  251.4877 244.0452 258.9303  58
## 46  243.2779 235.8278 250.7279  51
## 47  242.1050 234.3434 249.8666  50
## 48  259.6976 249.0607 270.3345  65
## 49  245.6235 238.5832 252.6639  53
## 50  231.5495 219.0579 244.0411  41
## 51  259.6976 249.0607 270.3345  65
## 52  235.0680 224.4153 245.7207  44
## 53  235.0680 224.4153 245.7207  44
## 54  253.8334 245.7101 261.9567  60
## 55  246.7964 239.8416 253.7512  54
## 56  242.1050 234.3434 249.8666  50
## 57  231.5495 219.0579 244.0411  41
## 58  246.7964 239.8416 253.7512  54
## 59  243.2779 235.8278 250.7279  51
## 60  243.2779 235.8278 250.7279  51
## 61  237.4137 227.8738 246.9536  46
## 62  251.4877 244.0452 258.9303  58
## 63  246.7964 239.8416 253.7512  54
## 64  246.7964 239.8416 253.7512  54
## 65  253.8334 245.7101 261.9567  60
## 66  253.8334 245.7101 261.9567  60
## 67  246.7964 239.8416 253.7512  54
```

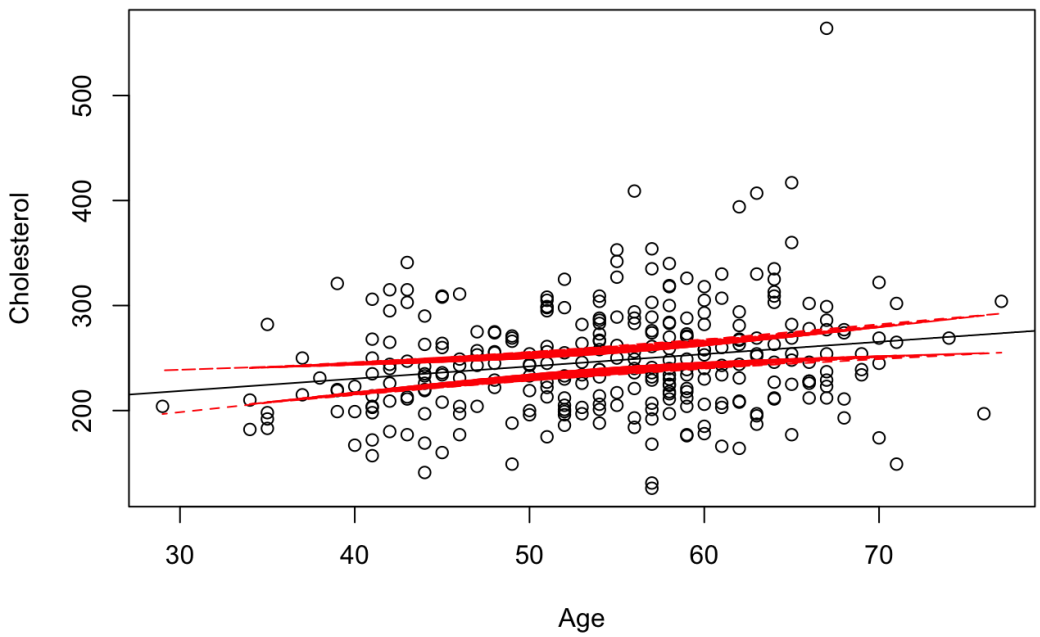
```
## 68  252.6606 244.9082 260.4129  59
## 69  237.4137 227.8738 246.9536  46
## 70  259.6976 249.0607 270.3345  65
## 71  262.0433 250.1996 273.8870  67
## 72  256.1791 247.1620 265.1961  62
## 73  259.6976 249.0607 270.3345  65
## 74  235.0680 224.4153 245.7207  44
## 75  259.6976 249.0607 270.3345  65
## 76  253.8334 245.7101 261.9567  60
## 77  243.2779 235.8278 250.7279  51
## 78  239.7594 231.1999 248.3188  48
## 79  251.4877 244.0452 258.9303  58
## 80  236.2408 226.1585 246.3232  45
## 81  245.6235 238.5832 252.6639  53
## 82  229.2038 215.4082 242.9994  39
## 83  263.2161 250.7418 275.6904  68
## 84  244.4507 237.2435 251.6579  52
## 85  235.0680 224.4153 245.7207  44
## 86  238.5865 229.5562 247.6168  47
## 88  245.6235 238.5832 252.6639  53
## 89  243.2779 235.8278 250.7279  51
## 90  260.8704 249.6402 272.1007  66
## 91  256.1791 247.1620 265.1961  62
## 92  256.1791 247.1620 265.1961  62
## 93  235.0680 224.4153 245.7207  44
## 94  257.3519 247.8262 266.8776  63
## 95  244.4507 237.2435 251.6579  52
## 96  252.6606 244.9082 260.4129  59
## 97  253.8334 245.7101 261.9567  60
## 98  244.4507 237.2435 251.6579  52
## 99  239.7594 231.1999 248.3188  48
## 100 236.2408 226.1585 246.3232  45
## 101 223.3396 206.1152 240.5641  34
## 102 250.3149 243.1132 257.5166  57
## 103 266.7346 252.2881 281.1811  71
## 104 240.9322 232.7981 249.0663  49
## 105 246.7964 239.8416 253.7512  54
## 106 252.6606 244.9082 260.4129  59
## 107 250.3149 243.1132 257.5166  57
## 108 255.0062 246.4589 263.5536  61
## 109 229.2038 215.4082 242.9994  39
## 110 255.0062 246.4589 263.5536  61
## 111 249.1421 242.1051 256.1790  56
## 112 244.4507 237.2435 251.6579  52
## 113 233.8952 222.6486 245.1417  43
## 114 256.1791 247.1620 265.1961  62
## 115 231.5495 219.0579 244.0411  41
## 116 251.4877 244.0452 258.9303  58
## 117 224.5125 207.9884 241.0366  35
## 118 257.3519 247.8262 266.8776  63
## 119 259.6976 249.0607 270.3345  65
## 120 239.7594 231.1999 248.3188  48
## 121 257.3519 247.8262 266.8776  63
## 122 243.2779 235.8278 250.7279  51
## 123 247.9692 241.0155 254.9229  55
## 124 259.6976 249.0607 270.3345  65
## 125 236.2408 226.1585 246.3232  45
## 126 249.1421 242.1051 256.1790  56
## 127 246.7964 239.8416 253.7512  54
## 128 235.0680 224.4153 245.7207  44
## 129 256.1791 247.1620 265.1961  62
## 130 246.7964 239.8416 253.7512  54
## 131 243.2779 235.8278 250.7279  51
## 132 217.4755 196.6773 238.2737  29
## 133 243.2779 235.8278 250.7279  51
## 134 233.8952 222.6486 245.1417  43
## 135 247.9692 241.0155 254.9229  55
## 136 265.5618 251.7841 279.3394  70
```

```
## 137 256.1791 247.1620 265.1961  62
## 138 224.5125 207.9884 241.0366  35
## 139 243.2779 235.8278 250.7279  51
## 140 252.6606 244.9082 260.4129  59
## 141 252.6606 244.9082 260.4129  59
## 142 244.4507 237.2435 251.6579  52
## 143 258.5248 248.4574 268.5921  64
## 144 251.4877 244.0452 258.9303  58
## 145 238.5865 229.5562 247.6168  47
## 146 250.3149 243.1132 257.5166  57
## 147 231.5495 219.0579 244.0411  41
## 148 236.2408 226.1585 246.3232  45
## 149 253.8334 245.7101 261.9567  60
## 150 244.4507 237.2435 251.6579  52
## 151 232.7223 220.8618 244.5829  42
## 152 262.0433 250.1996 273.8870  67
## 153 247.9692 241.0155 254.9229  55
## 154 258.5248 248.4574 268.5921  64
## 155 265.5618 251.7841 279.3394  70
## 156 243.2779 235.8278 250.7279  51
## 157 251.4877 244.0452 258.9303  58
## 158 253.8334 245.7101 261.9567  60
## 159 263.2161 250.7418 275.6904  68
## 160 237.4137 227.8738 246.9536  46
## 161 273.7716 255.1498 292.3935  77
## 162 246.7964 239.8416 253.7512  54
## 163 251.4877 244.0452 258.9303  58
## 164 239.7594 231.1999 248.3188  48
## 165 250.3149 243.1132 257.5166  57
## 167 246.7964 239.8416 253.7512  54
## 168 224.5125 207.9884 241.0366  35
## 169 236.2408 226.1585 246.3232  45
## 170 265.5618 251.7841 279.3394  70
## 171 245.6235 238.5832 252.6639  53
## 172 252.6606 244.9082 260.4129  59
## 173 256.1791 247.1620 265.1961  62
## 174 258.5248 248.4574 268.5921  64
## 175 250.3149 243.1132 257.5166  57
## 176 244.4507 237.2435 251.6579  52
## 177 249.1421 242.1051 256.1790  56
## 178 233.8952 222.6486 245.1417  43
## 179 245.6235 238.5832 252.6639  53
## 180 239.7594 231.1999 248.3188  48
## 181 249.1421 242.1051 256.1790  56
## 182 232.7223 220.8618 244.5829  42
## 183 252.6606 244.9082 260.4129  59
## 184 253.8334 245.7101 261.9567  60
## 185 257.3519 247.8262 266.8776  63
## 186 232.7223 220.8618 244.5829  42
## 187 260.8704 249.6402 272.1007  66
## 188 246.7964 239.8416 253.7512  54
## 189 264.3889 251.2693 277.5086  69
## 190 242.1050 234.3434 249.8666  50
## 191 243.2779 235.8278 250.7279  51
## 193 256.1791 247.1620 265.1961  62
## 194 263.2161 250.7418 275.6904  68
## 195 262.0433 250.1996 273.8870  67
## 196 264.3889 251.2693 277.5086  69
## 197 236.2408 226.1585 246.3232  45
## 198 242.1050 234.3434 249.8666  50
## 199 252.6606 244.9082 260.4129  59
## 200 242.1050 234.3434 249.8666  50
## 201 258.5248 248.4574 268.5921  64
## 202 250.3149 243.1132 257.5166  57
## 203 258.5248 248.4574 268.5921  64
## 204 233.8952 222.6486 245.1417  43
## 205 236.2408 226.1585 246.3232  45
## 206 251.4877 244.0452 258.9303  58
```

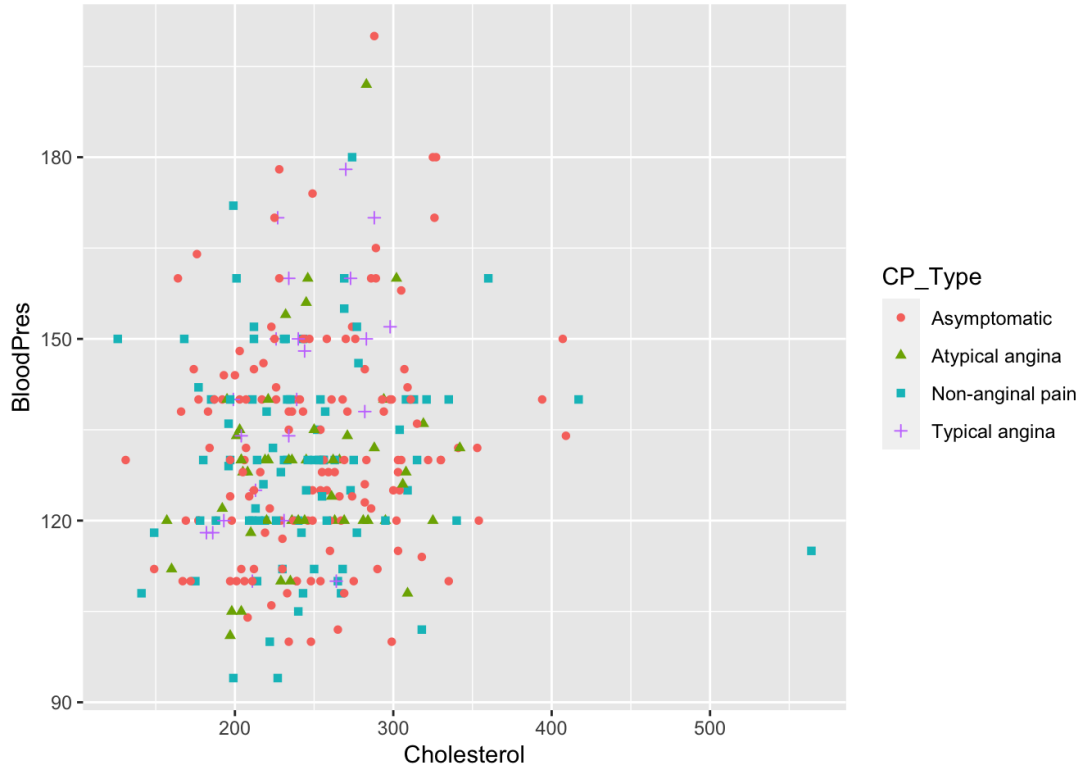
```
## 207 242.1050 234.3434 249.8666  50
## 208 247.9692 241.0155 254.9229  55
## 209 256.1791 247.1620 265.1961  62
## 210 226.8582 211.7148 242.0015  37
## 211 228.0310 213.5662 242.4957  38
## 212 231.5495 219.0579 244.0411  41
## 213 260.8704 249.6402 272.1007  66
## 214 244.4507 237.2435 251.6579  52
## 215 249.1421 242.1051 256.1790  56
## 216 237.4137 227.8738 246.9536  46
## 217 237.4137 227.8738 246.9536  46
## 218 258.5248 248.4574 268.5921  64
## 219 252.6606 244.9082 260.4129  59
## 220 231.5495 219.0579 244.0411  41
## 221 246.7964 239.8416 253.7512  54
## 222 229.2038 215.4082 242.9994  39
## 223 245.6235 238.5832 252.6639  53
## 224 257.3519 247.8262 266.8776  63
## 225 223.3396 206.1152 240.5641  34
## 226 238.5865 229.5562 247.6168  47
## 227 262.0433 250.1996 273.8870  67
## 228 246.7964 239.8416 253.7512  54
## 229 260.8704 249.6402 272.1007  66
## 230 244.4507 237.2435 251.6579  52
## 231 247.9692 241.0155 254.9229  55
## 232 240.9322 232.7981 249.0663  49
## 233 270.2531 253.7479 286.7584  74
## 234 246.7964 239.8416 253.7512  54
## 235 246.7964 239.8416 253.7512  54
## 236 249.1421 242.1051 256.1790  56
## 237 237.4137 227.8738 246.9536  46
## 238 240.9322 232.7981 249.0663  49
## 239 232.7223 220.8618 244.5829  42
## 240 231.5495 219.0579 244.0411  41
## 241 231.5495 219.0579 244.0411  41
## 242 240.9322 232.7981 249.0663  49
## 243 255.0062 246.4589 263.5536  61
## 244 253.8334 245.7101 261.9567  60
## 245 262.0433 250.1996 273.8870  67
## 246 251.4877 244.0452 258.9303  58
## 247 238.5865 229.5562 247.6168  47
## 248 244.4507 237.2435 251.6579  52
## 249 256.1791 247.1620 265.1961  62
## 250 250.3149 243.1132 257.5166  57
## 251 251.4877 244.0452 258.9303  58
## 252 258.5248 248.4574 268.5921  64
## 253 243.2779 235.8278 250.7279  51
## 254 233.8952 222.6486 245.1417  43
## 255 232.7223 220.8618 244.5829  42
## 256 262.0433 250.1996 273.8870  67
## 257 272.5988 254.6876 290.5099  76
## 258 265.5618 251.7841 279.3394  70
## 259 250.3149 243.1132 257.5166  57
## 260 235.0680 224.4153 245.7207  44
## 261 251.4877 244.0452 258.9303  58
## 262 253.8334 245.7101 261.9567  60
## 263 235.0680 224.4153 245.7207  44
## 264 255.0062 246.4589 263.5536  61
## 265 232.7223 220.8618 244.5829  42
## 267 252.6606 244.9082 260.4129  59
## 268 230.3767 217.2394 243.5140  40
## 269 232.7223 220.8618 244.5829  42
## 270 255.0062 246.4589 263.5536  61
## 271 260.8704 249.6402 272.1007  66
## 272 237.4137 227.8738 246.9536  46
## 273 266.7346 252.2881 281.1811  71
## 274 252.6606 244.9082 260.4129  59
## 275 258.5248 248.4574 268.5921  64
```

```
## 276 260.8704 249.6402 272.1007  66
## 277 229.2038 215.4082 242.9994  39
## 278 250.3149 243.1132 257.5166  57
## 279 251.4877 244.0452 258.9303  58
## 280 250.3149 243.1132 257.5166  57
## 281 238.5865 229.5562 247.6168  47
## 282 247.9692 241.0155 254.9229  55
## 283 224.5125 207.9884 241.0366  35
## 284 255.0062 246.4589 263.5536  61
## 285 251.4877 244.0452 258.9303  58
## 286 251.4877 244.0452 258.9303  58
## 288 249.1421 242.1051 256.1790  56
## 289 249.1421 242.1051 256.1790  56
## 290 262.0433 250.1996 273.8870  67
## 291 247.9692 241.0155 254.9229  55
## 292 235.0680 224.4153 245.7207  44
## 293 257.3519 247.8262 266.8776  63
## 294 257.3519 247.8262 266.8776  63
## 295 231.5495 219.0579 244.0411  41
## 296 252.6606 244.9082 260.4129  59
## 297 250.3149 243.1132 257.5166  57
## 298 236.2408 226.1585 246.3232  45
## 299 263.2161 250.7418 275.6904  68
## 300 250.3149 243.1132 257.5166  57
## 301 250.3149 243.1132 257.5166  57
```

```
plot(Cholesterol~Age, data = my_data)
abline(lm(Cholesterol~Age, data = my_data))
lines(prediction$Age, prediction$lwr, col="red",lty=2)
lines(prediction$Age, prediction$upr, col="red",lty=2)
```

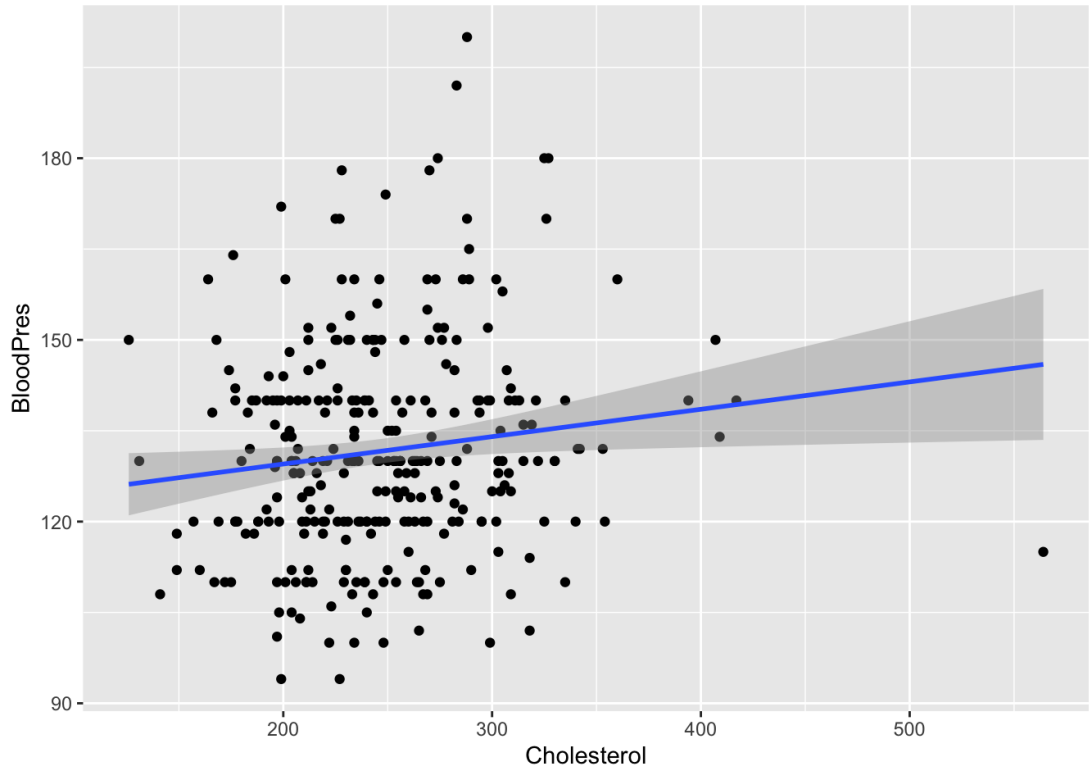



```
ggplot(data = my_data)+geom_point(mapping = aes(x=Cholesterol, y=BloodPres, color=CP_Type, shape=CP_Type))
```




```
my_data %>% ggplot(aes(x=Cholesterol, y=BloodPres))+geom_point()+
  stat_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# Linear Regression Model
linear_model2 <- lm(data = my_data, BloodPres~Cholesterol)
summary(linear_model2)
```

```
##
## Call:
## lm(formula = BloodPres ~ Cholesterol, data = my_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.727 -11.334  -1.992  10.004  66.517
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 120.47000    4.98815   24.15   <2e-16 ***
## Cholesterol   0.04518    0.01973    2.29   0.0227 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.65 on 294 degrees of freedom
## Multiple R-squared:  0.01752,    Adjusted R-squared:  0.01418
## F-statistic: 5.244 on 1 and 294 DF,  p-value: 0.02273
```

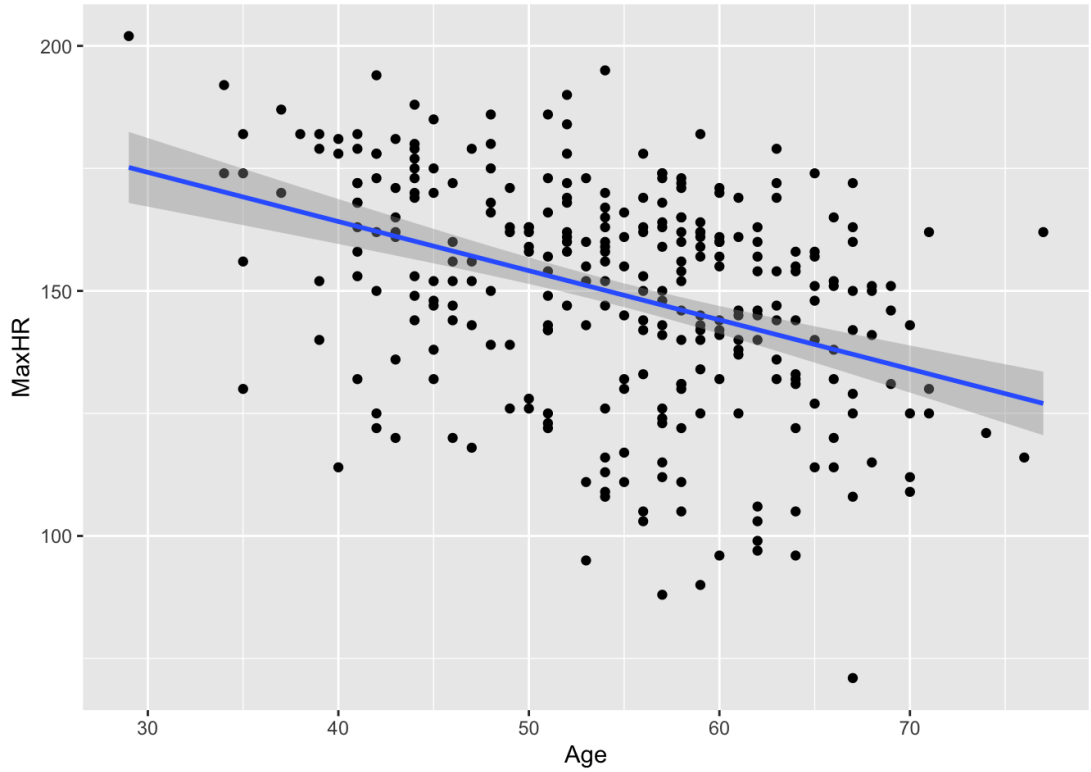
```
# run correlation
cor.test(my_data$BloodPres, my_data$Cholesterol)
```

```
##
##  Pearson's product-moment correlation
##
## data:  my_data$BloodPres and my_data$Cholesterol
## t = 2.29, df = 294, p-value = 0.02273
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.0186566 0.2427210
## sample estimates:
##       cor
## 0.1323796
```

Report: There was a significant relationship between cholesterol level and blood pressure(linear regression, r^2=0.01752, F_1,294=5.244, p=0.02273). The correlation between these two varibales was positive(r=0.1323796)

```
my_data %>% ggplot(aes(x=Age, y=MaxHR))+geom_point()+
  stat_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# Linear model for maximum heart rate and cholesterol
linear_model3 <- lm(data = my_data, MaxHR~Age)
summary(linear_model3)
```

```
##
## Call:
## lm(formula = MaxHR ~ Age, data = my_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -66.07 -12.07   3.88  15.85  44.89
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  204.294      7.516  27.182  < 2e-16 ***
## Age           -1.003      0.136  -7.377 1.66e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.15 on 294 degrees of freedom
## Multiple R-squared:  0.1562, Adjusted R-squared:  0.1533
## F-statistic: 54.42 on 1 and 294 DF,  p-value: 1.664e-12
```

```
# run correlation test
cor.test(my_data$MaxHR, my_data$Age)
```

```
##
##  Pearson's product-moment correlation
##
## data:  my_data$MaxHR and my_data$Age
## t = -7.3769, df = 294, p-value = 1.664e-12
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4872552 -0.2944666
## sample estimates:
##       cor
## -0.3952039
```

Report: There was a statistically significant relationship age and maximum heart rate(linear regression, r^2=0.1562, F_1,294=54.42, p=1.664e-12). The correlation between these two variables was negative(r=-0.3952039)

Machine Learning Model

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.1.2
```

```
## randomForest 4.7-1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(xgboost)
```

```
##
## Attaching package: 'xgboost'
```

```
## The following object is masked from 'package:dplyr':
##
##     slice
```

```
# data partition/ split data into training and testing
set.seed(123)
data_index <- createDataPartition(my_data$HD, p = .8, list = FALSE)
data_train <-my_data[data_index,]
data_test <- my_data[-data_index,]
```

Random Forest machine learning model

```
rf_model <- train(HD ~.,
                  data= data_test,
                  method='rf',
                  metric='Accuracy',
                  trControl=trainControl(method = 'cv', number = 10))
```

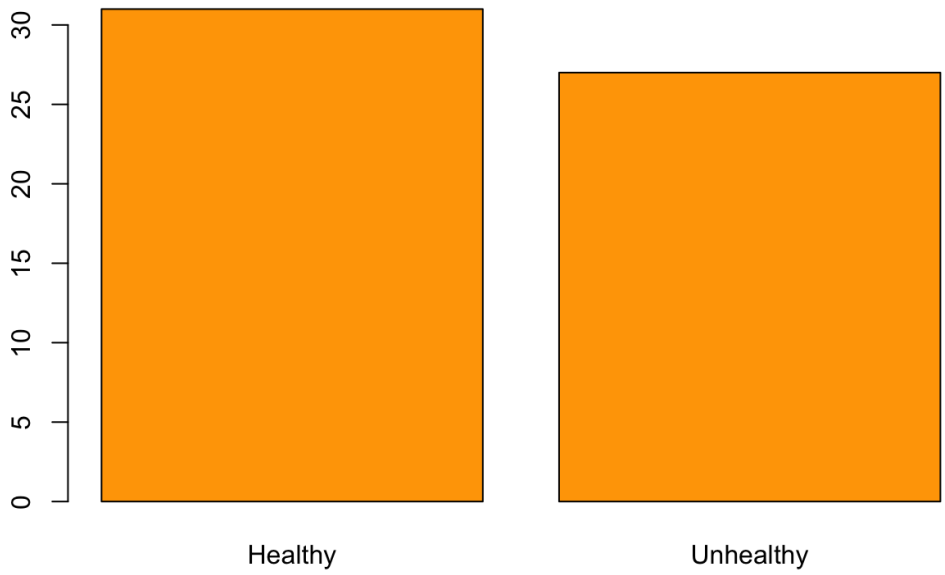
```
rf_model
```

```
## Random Forest
##
## 58 samples
## 13 predictors
##  2 classes: 'Healthy', 'Unhealthy'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 52, 52, 52, 52, 52, 52, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.7966667  0.5865385
##   11    0.7466667  0.4865385
##   20    0.7833333  0.5583333
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

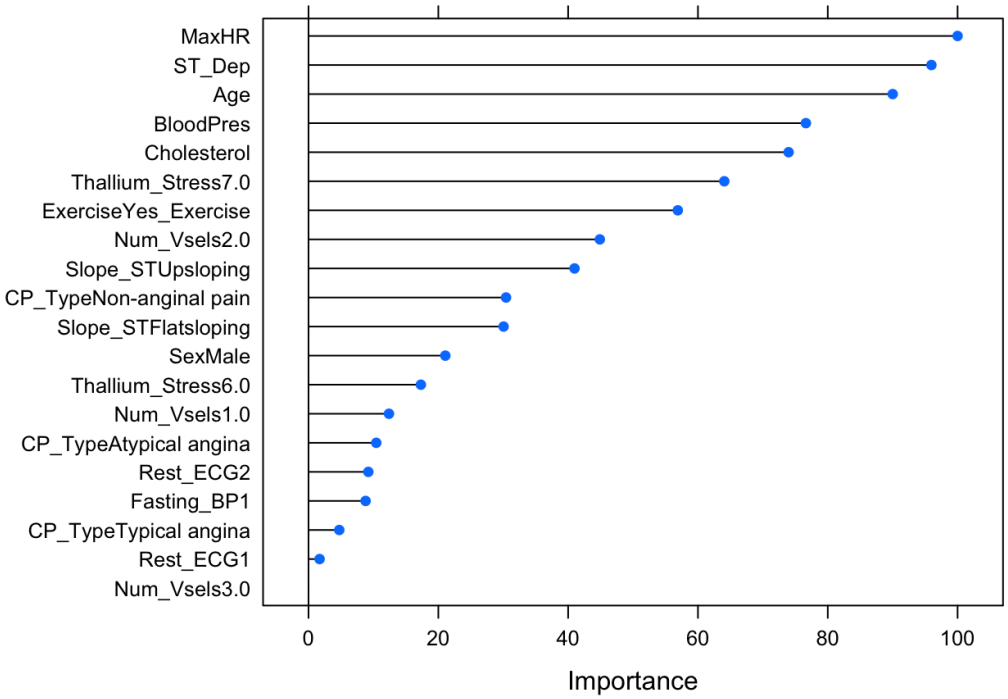
```
rf_preds <- predict(rf_model, data_test)
table(rf_preds)
```

```
## rf_preds
##   Healthy Unhealthy
##        31        27
```

```
plot(rf_preds, col='orange')
```




```
plot(varImp(rf_model, scale = TRUE))
```



Maximum heart rate is the best indicator for predicting heart condition of patients in this dataset as it shown in this visualization.

```
confusionMatrix(rf_preds, as.factor(data_test$HD))
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Healthy Unhealthy
##   Healthy        31         0
##   Unhealthy       0        27
##
##                Accuracy : 1
##                  95% CI : (0.9384, 1)
##     No Information Rate : 0.5345
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5345
##          Detection Rate : 0.5345
##    Detection Prevalence : 0.5345
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : Healthy
##
```