```r
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.5.2
```

```
## Warning: package 'stringr' was built under R version 4.5.2
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.1     v stringr   1.5.2
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.5.2
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.5.2
```

```r
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 4.5.2
```

```r
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.5.2
```

```
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
##
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.5.2
```

```
## corrplot 0.95 loaded
```

```r
library(janitor)
```

```
## Warning: package 'janitor' was built under R version 4.5.2
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```r
set.seed(123)
```

```r
getwd()
```

```
## [1] "C:/Users/charh/Desktop"
```

#Load the data

```r
heart <- read.table("processed.cleveland.data",
                    sep = ",",
                    header = FALSE,
                    na.strings = "?")
```

#Give data column names

```r
colnames(heart) <- c(
  "age", "sex", "cp", "trestbps", "chol",
  "fbs", "restecg", "thalach", "exang",
  "oldpeak", "slope", "ca", "thal", "num"
)
```

#Convert target variable to binary

```r
heart <- heart %>%
  mutate(
    disease = ifelse(num > 0, 1, 0),
    disease = factor(disease)
  )
```

#Drop multi-class col

```r
heart$num <- NULL
```

```r
glimpse(heart)
```

```
## Rows: 303
## Columns: 14
## $ age      <dbl> 63, 67, 67, 37, 41, 56, 62, 57, 63, 53, 57, 56, 56, 44, 52, 5~
## $ sex      <dbl> 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1~
## $ cp       <dbl> 1, 4, 4, 3, 2, 2, 4, 4, 4, 4, 4, 2, 3, 2, 3, 3, 2, 4, 3, 2, 1~
## $ trestbps <dbl> 145, 160, 120, 130, 130, 120, 140, 120, 130, 140, 140, 140, 1~
## $ chol     <dbl> 233, 286, 229, 250, 204, 236, 268, 354, 254, 203, 192, 294, 2~
## $ fbs      <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0~
## $ restecg  <dbl> 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 0, 2, 2, 0, 0, 0, 0, 0, 0, 0, 2~
## $ thalach  <dbl> 150, 108, 129, 187, 172, 178, 160, 163, 147, 155, 148, 153, 1~
## $ exang    <dbl> 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1~
## $ oldpeak  <dbl> 2.3, 1.5, 2.6, 3.5, 1.4, 0.8, 3.6, 0.6, 1.4, 3.1, 0.4, 1.3, 0~
## $ slope    <dbl> 3, 2, 2, 3, 1, 1, 3, 1, 2, 3, 2, 2, 2, 1, 1, 1, 3, 1, 1, 1, 2~
## $ ca       <dbl> 0, 3, 2, 0, 0, 0, 2, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0~
## $ thal     <dbl> 6, 3, 7, 3, 3, 3, 3, 3, 7, 7, 6, 3, 6, 7, 7, 3, 7, 3, 3, 3, 3~
## $ disease  <fct> 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0~
```

```r
colSums(is.na(heart))
```

```
##      age      sex       cp trestbps     chol      fbs  restecg  thalach
##        0        0        0        0        0        0        0        0
##    exang  oldpeak    slope       ca     thal  disease
##        0        0        0        4        2        0
```

#Median imputation for numeric #Mode for categorical

```r
for (col in names(heart)) {
  if (is.numeric(heart[[col]])) {
    heart[[col]][is.na(heart[[col]])] <- median(heart[[col]], na.rm = TRUE)
  } else {
    heart[[col]][is.na(heart[[col]])] <- names(which.max(table(heart[[col]])))
  }
}
```

#Summary Stats

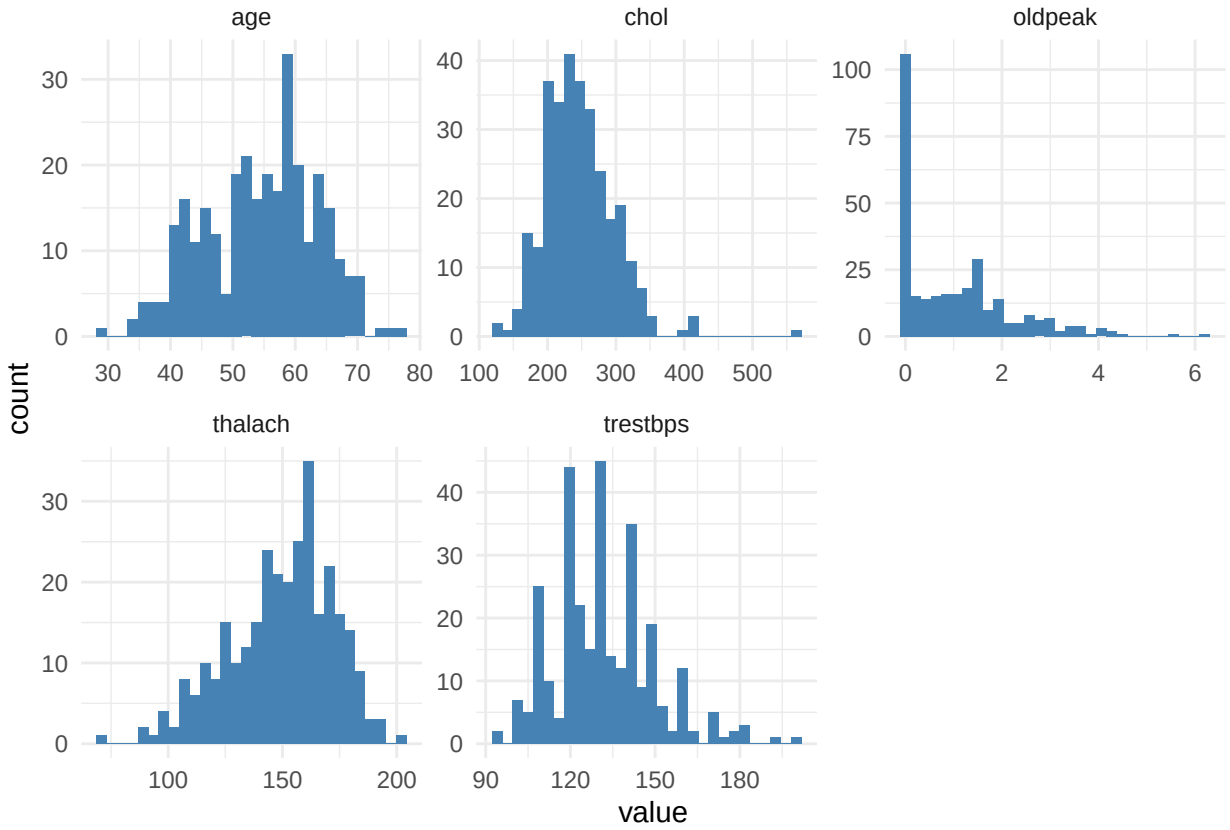
```r
summary(heart)
```

```
##       age             sex               cp           trestbps
##  Min.   :29.00   Min.   :0.0000   Min.   :1.000   Min.   : 94.0
##  1st Qu.:48.00   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:120.0
##  Median :56.00   Median :1.0000   Median :3.000   Median :130.0
##  Mean   :54.44   Mean   :0.6799   Mean   :3.158   Mean   :131.7
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :4.000   Max.   :200.0
##       chol            fbs            restecg          thalach
##  Min.   :126.0   Min.   :0.0000   Min.   :0.0000   Min.   : 71.0
```

```
##  1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:133.5
##  Median :241.0   Median :0.0000   Median :1.0000   Median :153.0
##  Mean   :246.7   Mean   :0.1485   Mean   :0.9901   Mean   :149.6
##  3rd Qu.:275.0   3rd Qu.:0.0000   3rd Qu.:2.0000   3rd Qu.:166.0
##  Max.   :564.0   Max.   :1.0000   Max.   :2.0000   Max.   :202.0
##      exang           oldpeak         slope             ca
##  Min.   :0.0000   Min.   :0.00   Min.   :1.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000
##  Median :0.0000   Median :0.80   Median :2.000   Median :0.0000
##  Mean   :0.3267   Mean   :1.04   Mean   :1.601   Mean   :0.6634
##  3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :6.20   Max.   :3.000   Max.   :3.0000
##       thal       disease
##  Min.   :3.000   0:164
##  1st Qu.:3.000   1:139
##  Median :3.000
##  Mean   :4.723
##  3rd Qu.:7.000
##  Max.   :7.000
```

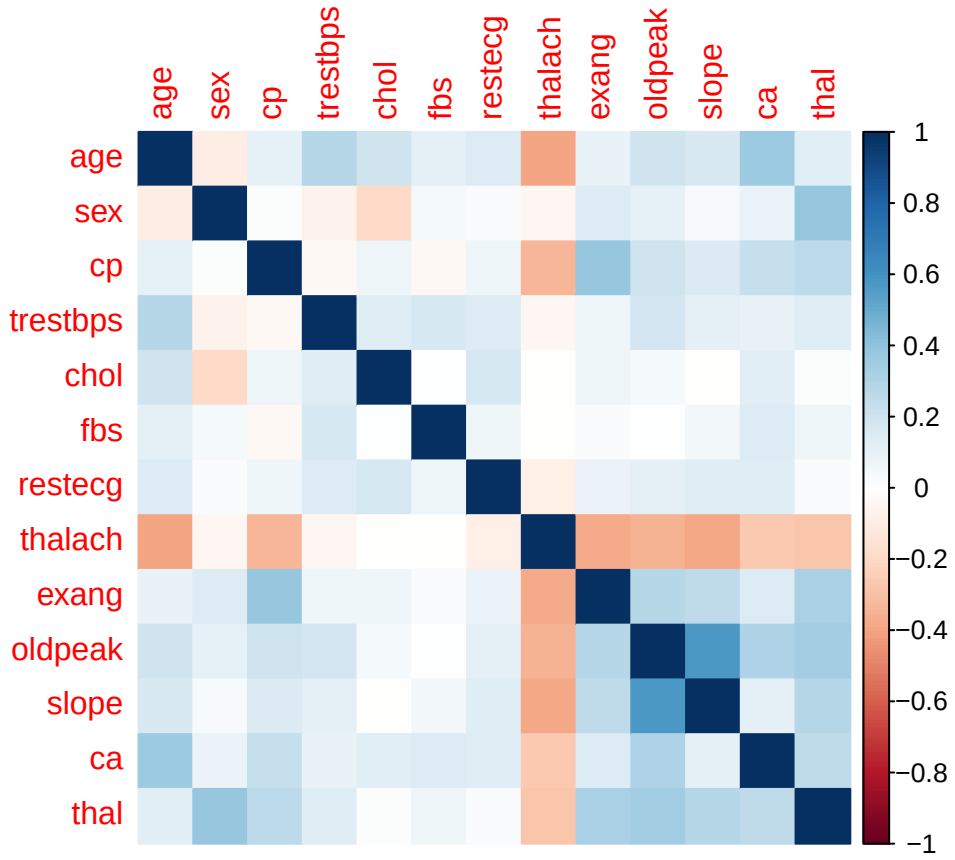
#Histogram of numeric colums

```
heart %>%
  select(age, trestbps, chol, thalach, oldpeak) %>%
  gather() %>%
  ggplot(aes(value)) +
  geom_histogram(bins = 30, fill = "steelblue") +
  facet_wrap(~ key, scales = "free") +
  theme_minimal()
```



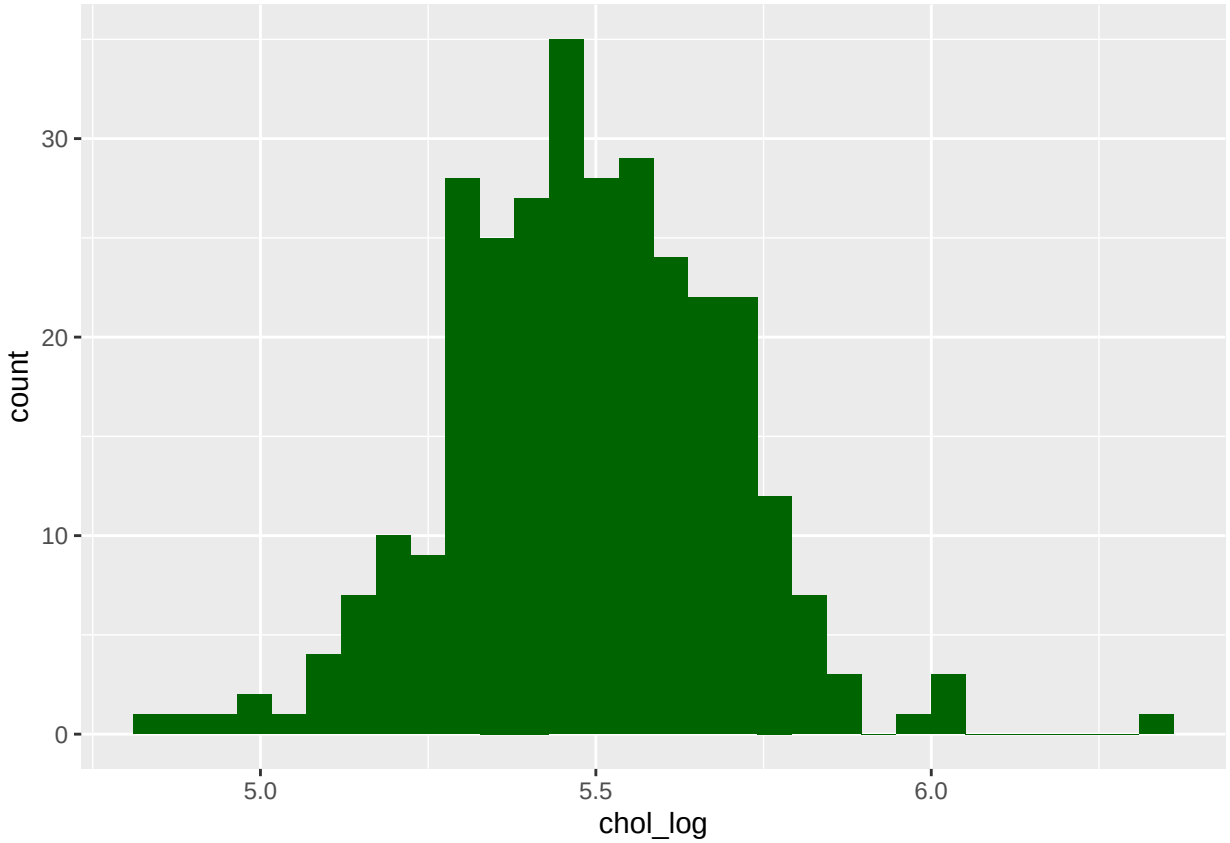
#Correlation plot

```
numeric_vars <- heart %>% select_if(is.numeric)
corrplot(cor(numeric_vars), method = "color")
```

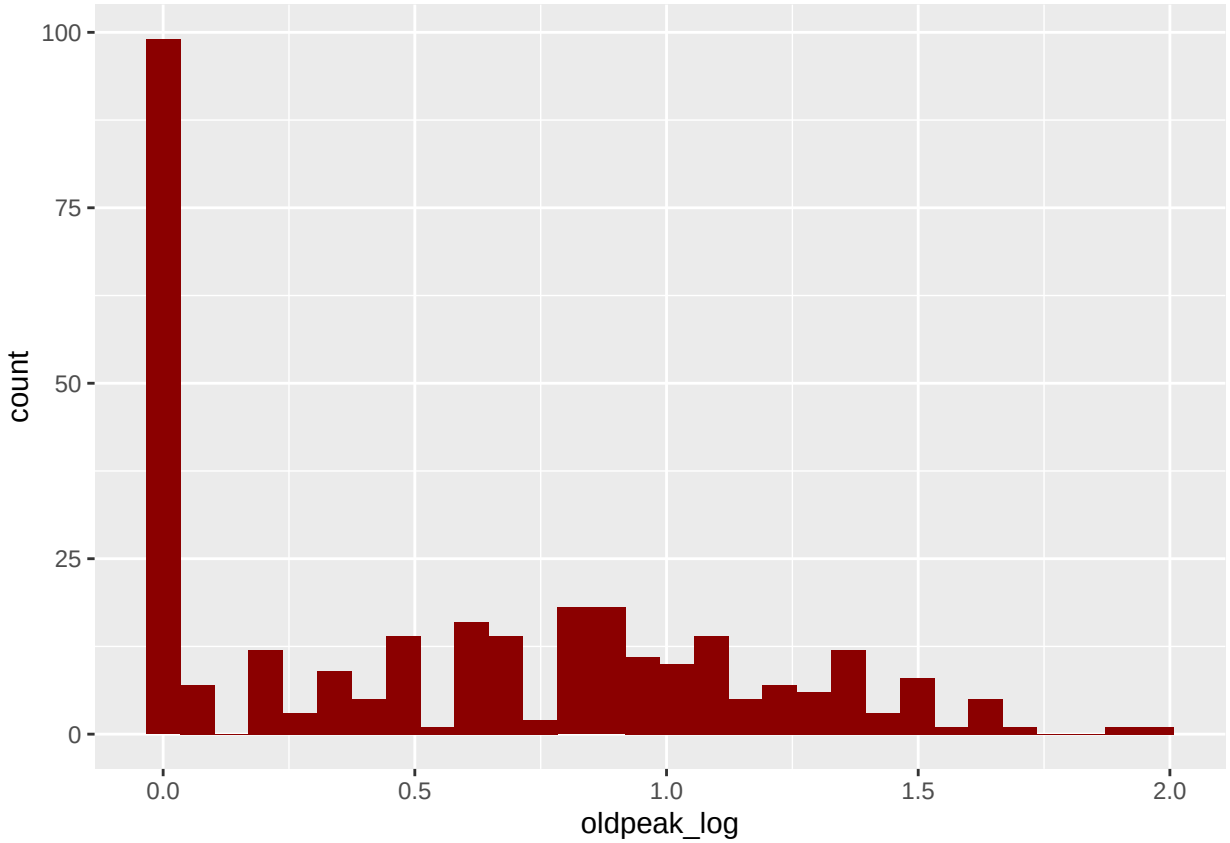



```
heart <- heart %>%
  mutate(
    chol_log = log(chol),
    oldpeak_log = log(oldpeak + 1)  # avoid log(0)
  )

#Check transformed distributions
ggplot(heart, aes(chol_log)) + geom_histogram(bins = 30, fill = "darkgreen")
```




```
ggplot(heart, aes(oldpeak_log)) + geom_histogram(bins = 30, fill = "darkred")
```




```
train_index <- createDataPartition(heart$disease, p = 0.8, list = FALSE)
train <- heart[train_index, ]
test  <- heart[-train_index, ]
```

```
log_model <- glm(disease ~ age + sex + cp + trestbps + chol_log +
                   thalach + exang + oldpeak_log + slope + ca + thal,
                 data = train,
                 family = binomial)

summary(log_model)
```

```
##
## Call:
## glm(formula = disease ~ age + sex + cp + trestbps + chol_log +
##     thalach + exang + oldpeak_log + slope + ca + thal, family = binomial,
##     data = train)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.720695   6.223980  -1.722 0.084982 .
## age          -0.009907   0.026072  -0.380 0.703957
## sex           1.617671   0.535150   3.023 0.002504 **
## cp            0.596877   0.211163   2.827 0.004704 **
## trestbps      0.021673   0.011497   1.885 0.059419 .
## chol_log      1.037735   1.063548   0.976 0.329199
```

```
## thalach      -0.025793   0.011728  -2.199 0.027862 *
## exang         1.078295   0.443475   2.431 0.015038 *
## oldpeak_log   0.536189   0.487583   1.100 0.271469
## slope         0.489037   0.373771   1.308 0.190742
## ca            0.985229   0.257613   3.824 0.000131 ***
## thal          0.279325   0.105258   2.654 0.007961 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 336.61  on 243  degrees of freedom
## Residual deviance: 173.15  on 232  degrees of freedom
## AIC: 197.15
##
## Number of Fisher Scoring iterations: 6
```

```
exp(cbind(OR = coef(log_model), confint(log_model)))
```

```
## Waiting for profiling to be done...
```

```
##                        OR        2.5 %     97.5 %
## (Intercept) 2.208316e-05 8.778553e-11  4.2033738
## age         9.901421e-01 9.402760e-01  1.0420613
## sex         5.041335e+00 1.827108e+00 15.1094670
## cp          1.816438e+00 1.212642e+00  2.7931559
## trestbps    1.021910e+00 9.993478e-01  1.0457683
## chol_log    2.822817e+00 3.521370e-01 23.7091671
## thalach     9.745369e-01 9.515017e-01  0.9966345
## exang       2.939662e+00 1.234067e+00  7.1005581
## oldpeak_log 1.709480e+00 6.582892e-01  4.5028508
## slope       1.630746e+00 7.760973e-01  3.3982154
## ca          2.678426e+00 1.657918e+00  4.5810995
## thal        1.322237e+00 1.076489e+00  1.6298324
```

```
#Predictions on test set
log_prob <- predict(log_model, test, type = "response")
log_pred <- ifelse(log_prob > 0.5, 1, 0)

#Confusion matrix
confusionMatrix(factor(log_pred), test$disease)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 28  5
##          1  4 22
##
##                Accuracy : 0.8475
##                  95% CI : (0.7301, 0.9278)
##     No Information Rate : 0.5424
```

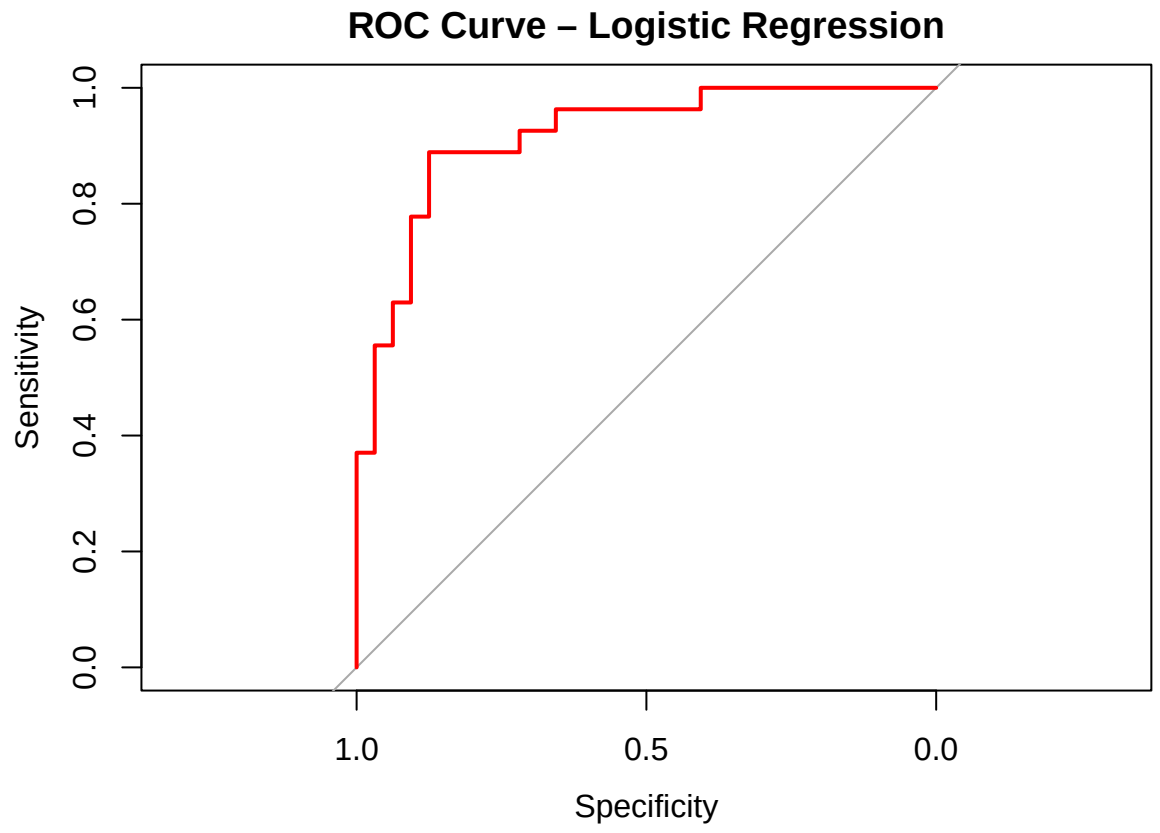
```
##     P-Value [Acc > NIR] : 7.195e-07
##
##                   Kappa : 0.6918
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8750
##             Specificity : 0.8148
##          Pos Pred Value : 0.8485
##          Neg Pred Value : 0.8462
##              Prevalence : 0.5424
##          Detection Rate : 0.4746
##    Detection Prevalence : 0.5593
##       Balanced Accuracy : 0.8449
##
##        'Positive' Class : 0
##
```

```r
#ROC curve & AUC
roc_obj <- roc(test$disease, log_prob)
```

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

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

```r
plot(roc_obj, col = "red", main = "ROC Curve - Logistic Regression")
```

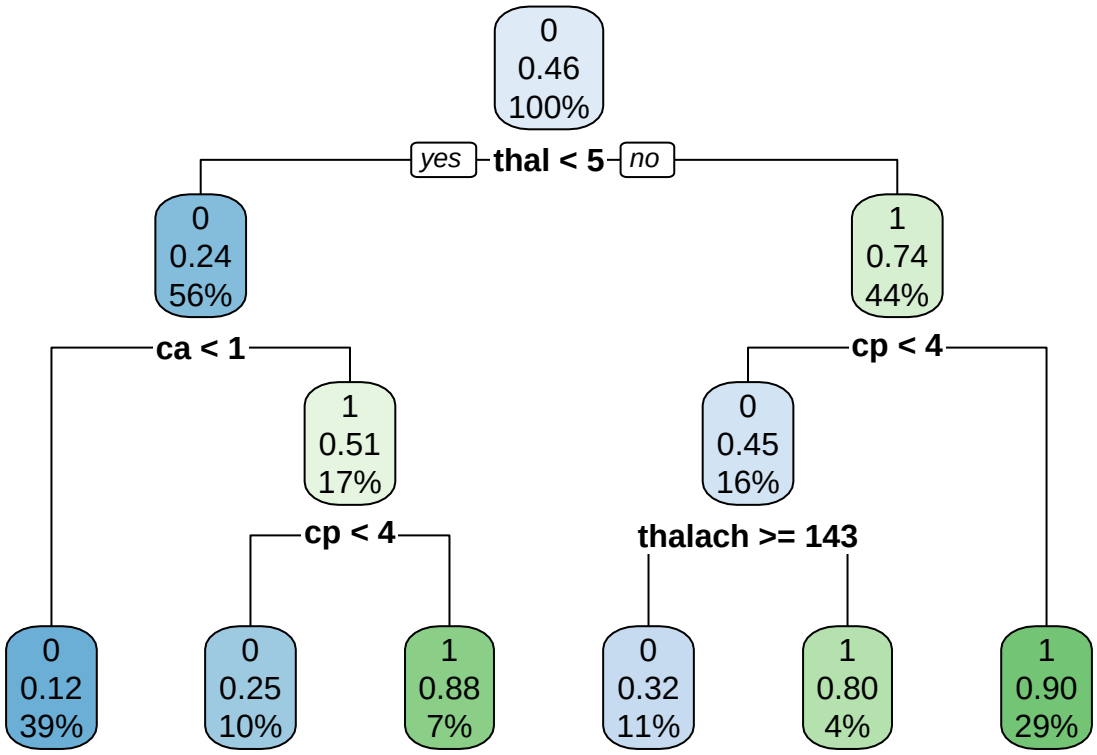


```r
auc(roc_obj)
```

```
## Area under the curve: 0.9167
```

```r
tree_model <- rpart(disease ~ age + sex + cp + trestbps + chol_log +
                      thalach + exang + oldpeak_log + slope + ca + thal,
                    data = train,
                    method = "class",
                    cp = 0.01)

#Visualize tree
rpart.plot(tree_model, type = 2, extra = 106)
```




```r
#Predictions & evaluation
tree_pred <- predict(tree_model, test, type = "class")
confusionMatrix(tree_pred, test$disease)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 28  6
##          1  4 21
##
```

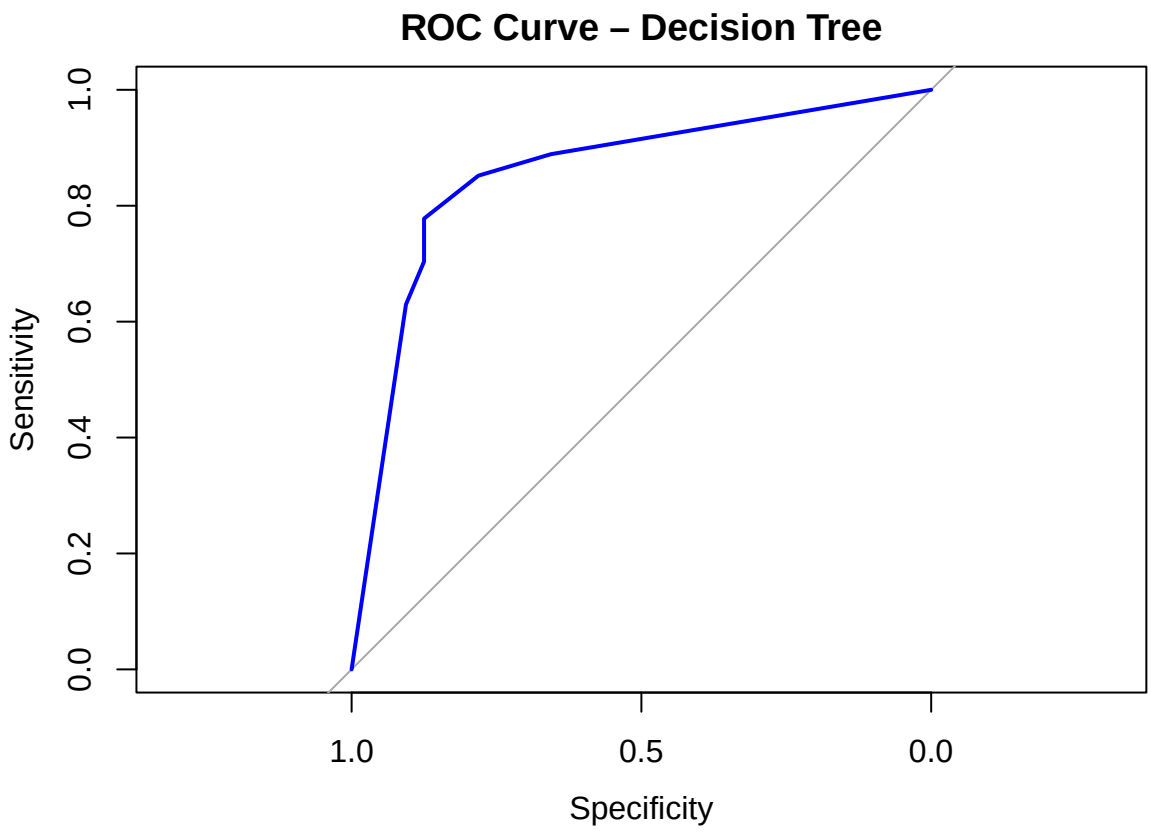
```
##                Accuracy : 0.8305
##                  95% CI : (0.7103, 0.9156)
##     No Information Rate : 0.5424
##     P-Value [Acc > NIR] : 3.14e-06
##
##                   Kappa : 0.6566
##
##  Mcnemar's Test P-Value : 0.7518
##
##             Sensitivity : 0.8750
##             Specificity : 0.7778
##          Pos Pred Value : 0.8235
##          Neg Pred Value : 0.8400
##              Prevalence : 0.5424
##          Detection Rate : 0.4746
##    Detection Prevalence : 0.5763
##       Balanced Accuracy : 0.8264
##
##        'Positive' Class : 0
##
```

```r
#ROC & AUC for tree
tree_prob <- predict(tree_model, test)[, 2]
roc_tree <- roc(test$disease, tree_prob)
```

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

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

```r
plot(roc_tree, col = "blue", main = "ROC Curve - Decision Tree")
```




```
auc(roc_tree)
```

```
## Area under the curve: 0.8553
```

```
train_control <- trainControl(
  method = "cv",
  number = 5,
  classProbs = TRUE,
  summaryFunction = twoClassSummary
)

heart$disease <- factor(heart$disease, levels = c(0, 1), labels = c("No", "Yes"))

# Check levels
levels(heart$disease)
```

```
## [1] "No"  "Yes"
```

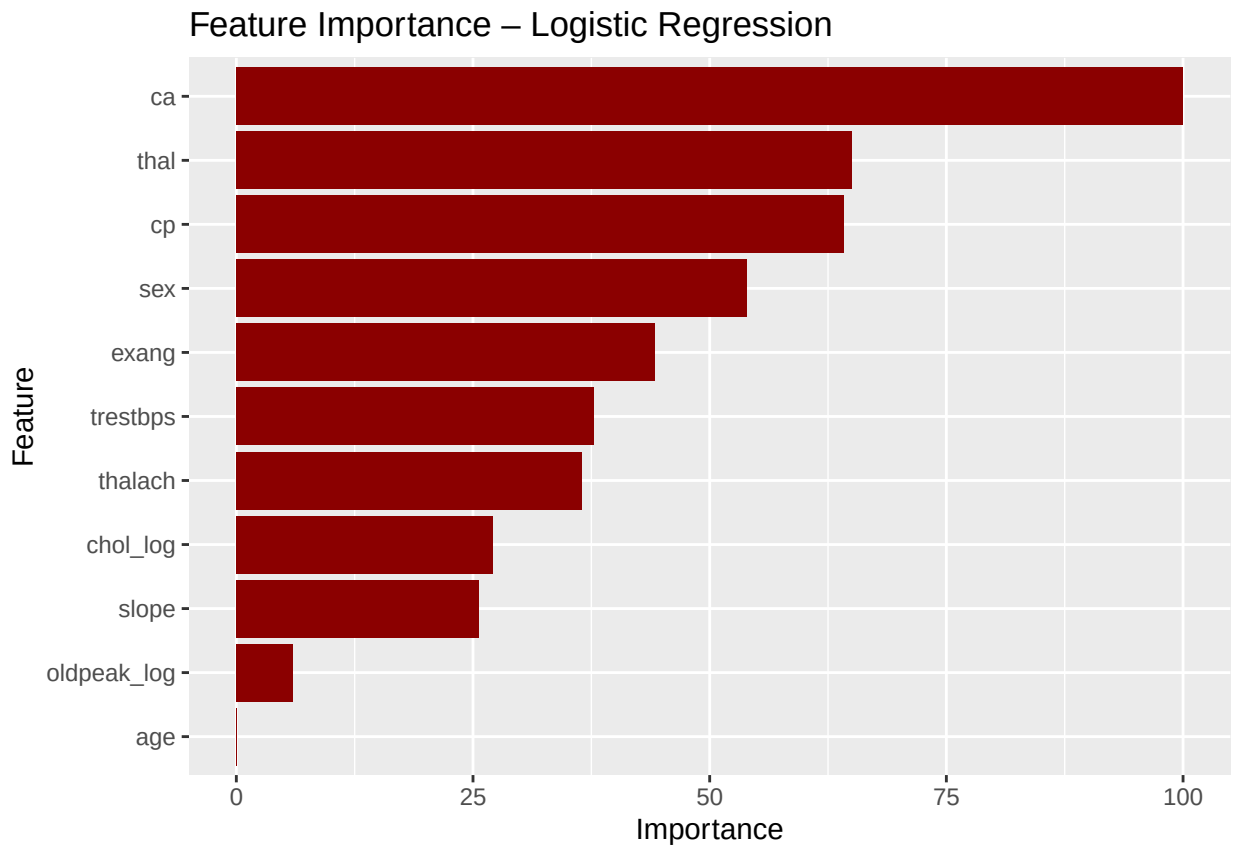
```
cv_log_model <- train(
  disease ~ age + sex + cp + trestbps + chol_log +
    thalach + exang + oldpeak_log + slope + ca + thal,
  data = heart,
  method = "glm",
  family = binomial,
```

```
  metric = "ROC",
  trControl = train_control
)

cv_log_model
```

```
## Generalized Linear Model
##
## 303 samples
##  11 predictor
##   2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 242, 242, 243, 243, 242
## Resampling results:
##
##   ROC        Sens       Spec
##   0.8996139  0.8784091  0.805291
```

```
varImp(cv_log_model) %>%
  ggplot(aes(x = Overall, y = reorder(rownames(.), Overall))) +
  geom_col(fill = "darkred") +
  labs(title = "Feature Importance - Logistic Regression")
```



```r
results <- data.frame(
  Model = c("Logistic Regression", "Decision Tree"),
  Accuracy = c(
    confusionMatrix(factor(log_pred), factor(test$disease))$overall["Accuracy"],
    confusionMatrix(tree_pred, test$disease)$overall["Accuracy"]
  ),
  AUC = c(
    auc(roc_obj),
    auc(roc_tree)
  )
)

results
```

```
##                 Model  Accuracy       AUC
## 1 Logistic Regression 0.8474576 0.9166667
## 2       Decision Tree 0.8305085 0.8553241
```