Project Two: Logistic Regression and Random Forests

For Project Two, you have been asked to create different models analyzing a Heart Disease data set. Before beginning work on the project, be sure to read through the Project Two Guidelines and Rubric to understand what you need to do and how you will be graded on this assignment. Be sure to carefully review the Project Two Summary Report template, which contains all of the questions that you will need to answer about the regression analyses you are performing.

For this project, you will be writing all the scripts yourself. You may reference the textbook and your previous work on the problem sets to help you write the scripts.

Scenario

You are a data analyst researching risk factors for heart disease at a university hospital. You have access to a large set of historical data that you can use to analyze patterns between different health indicators (e.g. fasting blood sugar, maximum heart rate, etc.) and the presence of heart disease. You have been asked to create different logistic regression models that predict whether or not a person is at risk for heart disease. A model like this could eventually be used to evaluate medical records and look for risks that might not be obvious to human doctors. You have also been asked to create a classification random forest model to predict the risk of heart disease and a regression random forest model to predict the maximum heart rate achieved.

There are several variables in this data set, but you will be working with the following important variables:

Variable	**What does it represent?**
age	The person's age in years
sex	The person's sex (1 = male, 0 = female)
cp	The type of chest pain experienced (0=no pain, 1=typical angina, 2=atypical angina, 3=non-anginal pain)
trestbps	The person's resting blood pressure

chol	The person's cholesterol measurement in mg/dl
fbs	The person's fasting blood sugar is greater than 120 mg/dl (1 = true, 0 = false)
restecg	Resting electrocardiographic measurement (0=normal, 1=having ST-T wave abnormality, 2=showing probable or definite left ventricular hypertrophy by Estes' criteria)
thalach	The person's maximum heart rate achieved
exang	Exercise-induced angina (1=yes, 0=no)
oldpeak	ST depression induced by exercise relative to rest ('ST' relates to positions on the ECG plot)
slope	The slope of the peak exercise ST segment (1=upsloping, 2=flat, 3=downsloping)
ca	The number of major vessels (0-3)
target	Heart disease (0=no, 1=yes)

Install Libraries

In the following code block, you will install appropriate libraries to use in this project.

Click the **Run** button on the toolbar to run this code.

In [1]:

```
install.packages("ResourceSelection")
install.packages("pROC")
install.packages("rpart.plot")
```

```
Installing package into '/home/codio/R/x86_64-pc-lin
ux-gnu-library/3.4'
(as 'lib' is unspecified)
Installing package into '/home/codio/R/x86_64-pc-lin
ux-gnu-library/3.4'
(as 'lib' is unspecified)
Installing package into '/home/codio/R/x86_64-pc-lin
ux-gnu-library/3.4'
(as 'lib' is unspecified)
```

Prepare Your Data Set

In the following code block, you have been given the R code to prepare your data set.

Click the **Run** button on the toolbar to run this code.

In [2]:

```
heart_data <- read.csv(file="heart_disease.csv", header=TRUE,
sep=",")

# Converting appropriate variables to factors
heart_data <- within(heart_data, {
   target <- factor(target)
   sex <- factor(sex)
   cp <- factor(cp)
   fbs <- factor(fbs)
   restecg <- factor(restecg)
   exang <- factor(exang)
   slope <- factor(slope)
   ca <- factor(ca)
   thal <- factor(thal)
})

head(heart_data, 10)

print("Number of variables")
ncol(heart_data)

print("Number of rows")
nrow(heart_data)
```

A data.frame: 10 × 14

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang
<int>	<fct>	<fct>	<int>	<int>	<fct>	<fct>	<int>	<fct>
62	1	2	130	231	0	1	146	0
58	0	0	130	197	0	1	131	0
60	0	3	150	240	0	1	171	0
63	1	0	140	187	0	0	144	1
62	1	0	120	267	0	1	99	1
63	0	2	135	252	0	0	172	0
43	1	0	150	247	0	1	171	0
42	1	2	120	240	1	1	194	0
59	1	2	126	218	1	1	134	0
48	1	0	124	274	0	0	166	0

```
[1] "Number of variables"
```

14

```
[1] "Number of rows"
```

303

Model #1 - First Logistic Regression Model

You have been asked to create a logistic regression model for heart disease *(target)* using the variables age *(age)*, resting blood pressure *(trestbps)*, and maximum heart rate achieved *(thalach)*. Before writing any code, review Section 3 of the Summary Report template to see the questions you will be answering about your logistic regression model.

Run your scripts to get the outputs of your regression analysis. Then use the outputs to answer the questions in your summary report.

Note: Use the + (plus) button to add new code blocks, if needed.

In [3]:

```
logit_one <- glm(target ~ age + trestbps + thalach, data=heart_d
ata, family = "binomial")
summary(logit_one)
```

```
Call:
glm(formula = target ~ age + trestbps + thalach, fam
ily = "binomial",
    data = heart_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0257  -1.0069   0.5688   0.9203   2.0476

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.576198   1.633928  -2.189   0.0286 *
age         -0.009424   0.016080  -0.586   0.5578
trestbps    -0.016019   0.007767  -2.063   0.0392 *
thalach      0.042697   0.006950   6.144 8.06e-10 **
*
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
0.1 ' ' 1

(Dispersion parameter for binomial family taken to b
e 1)

    Null deviance: 417.64  on 302  degrees of freedo
m
Residual deviance: 353.28  on 299  degrees of freedo
m
AIC: 361.28

Number of Fisher Scoring iterations: 3
```

In [4]:

```
# Hosmer-Lemeshow GOF
library(ResourceSelection)

print("Hosmer-Lemeshow Goodness of Fit Test")
hl = hoslem.test(logit_one$y, fitted(logit_one), g=50)
hl

# Wald Test
conf_int_one <- confint.default(logit_one, level=0.95)
round(conf_int_one,4)
```

```
ResourceSelection 0.3-5          2019-07-22

[1] "Hosmer-Lemeshow Goodness of Fit Test"

        Hosmer and Lemeshow goodness of fit (GOF) te
st

data:  logit_one$y, fitted(logit_one)
X-squared = 41.978, df = 48, p-value = 0.7168
```

A matrix: 4 × 2 of type dbl

	2.5 %	**97.5 %**
(Intercept)	-6.7786	-0.3738
age	-0.0409	0.0221
trestbps	-0.0312	-0.0008
thalach	0.0291	0.0563

In [6]:

```
# confusion matrix
# Predict target yes or no for the data set using the model
matrix_model_one <- heart_data[c('age', 'trestbps', 'thalach')]
pred <- predict(logit_one, newdata=matrix_model_one, type='respo
nse')

depvar_pred = as.factor(ifelse(pred >= 0.5, '1', '0'))

# This creates the confusion matrix
conf.matrix <- table(heart_data$target, depvar_pred)[c('0','1'),
c('0','1')]
rownames(conf.matrix) <- paste("Actual", rownames(conf.matrix),
sep = ": Target=")
colnames(conf.matrix) <- paste("Prediction", colnames(conf.matri
x), sep = ": Target=")

# Nicely formatted confusion matrix
print("Confusion Matrix")
format(conf.matrix,justify="centre",digit=2)
```

```
[1] "Confusion Matrix"
```

A matrix: 2 × 2 of type chr

	Prediction: Target=0	**Prediction: Target=1**
Actual: Target=0	83	55
Actual: Target=1	38	127

In [7]:

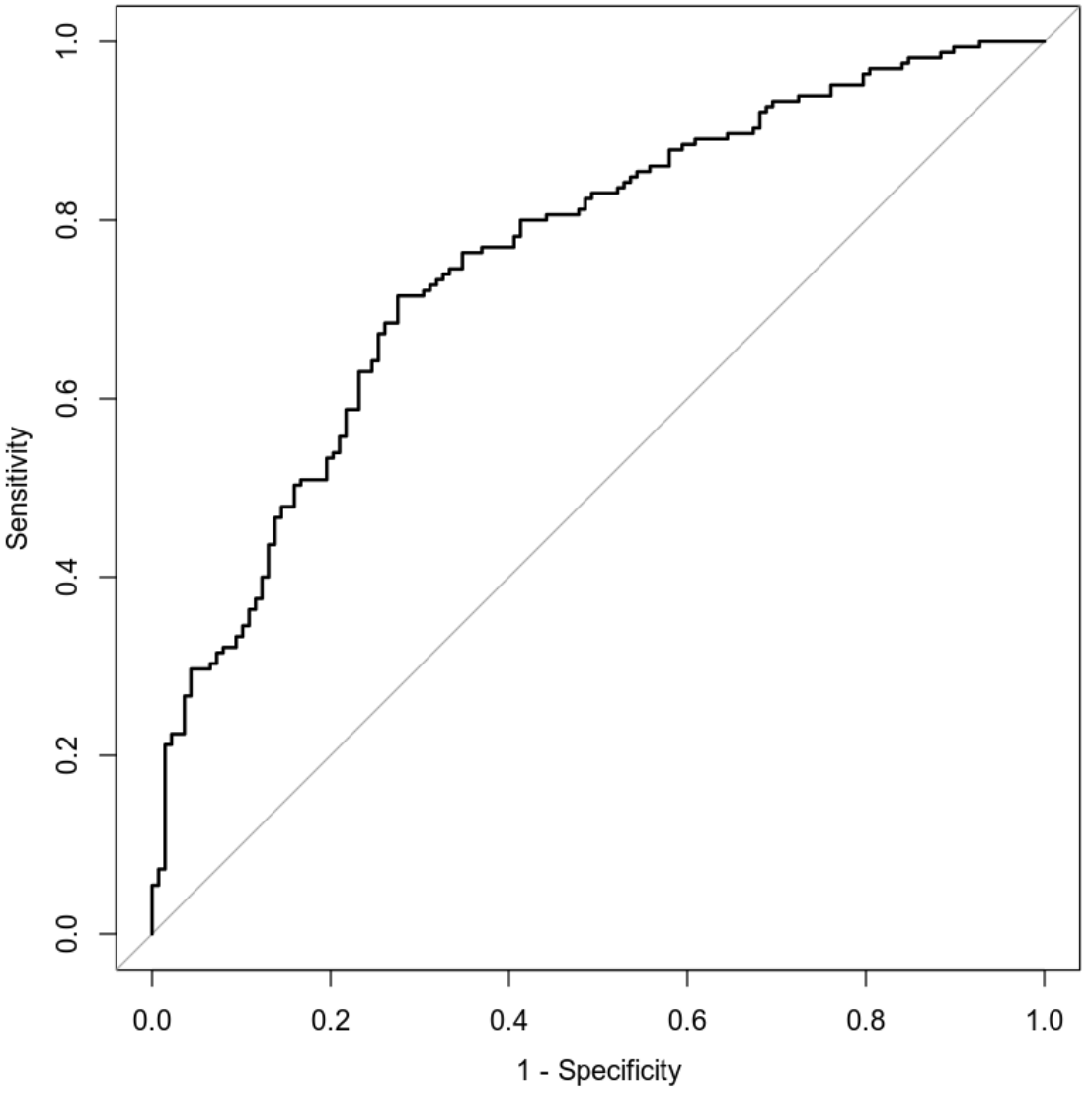
```
library(pROC)

labels <- heart_data$target
predictions <- logit_one$fitted.values

roc <- roc(labels ~ predictions)

print("Area Under the Curve (AUC)")
round(auc(roc),4)

print("ROC Curve")
# True Positive Rate (Sensitivity) and False Positive Rate (1 - Specificity)
plot(roc, legacy.axes = TRUE)
```

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

Attaching package: 'pROC'

The following objects are masked from 'package:stats
':

    cov, smooth, var

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

```
[1] "Area Under the Curve (AUC)"
```

0.7575

```
[1] "ROC Curve"
```


Sensitivity
1 - Specificity
0.0
0.2
0.4
0.6
0.8
1.0

In [8]:

```
print("Prediction: Age = 50, resting blood pressure = 122, max h
eart rate = 140")
newdata_one <- data.frame(age=50, trestbps=122, thalach=140)
pred1 <- predict(logit_one, newdata_one, type='response')
round(pred1, 4)


print("Prediction: Age = 50, resting blood pressure = 140, max h
eart rate = 170")
newdata_two <- data.frame(age=50, trestbps=140, thalach=170)
pred2 <- predict(logit_one, newdata_two, type='response')
round(pred2, 4)


odds1 <- ((pred1) /(1 - pred1))
print("Odds for first prediction")
round(odds1, 4)

odds2 <- ((pred2) /(1 - pred2))
print("Odds for second prediction")
round(odds2, 4)
```

```
[1] "Prediction: Age = 50, resting blood pressure =
122, max heart rate = 140"
```

1: 0.4939

```
[1] "Prediction: Age = 50, resting blood pressure =
140, max heart rate = 170"
```

1: 0.7248

```
[1] "Odds for first prediction"
```

1: 0.9761

```
[1] "Odds for second prediction"
```

1: 2.6335

Model #2 - Second Logistic Regression Model

You have been asked to create a logistic regression model for heart disease *(target)* using the variables maximum heart rate achieved *(thalach)*, age of the individual *(age)*, sex of the individual *(sex)*, exercise-induced angina *(exang)*, and type of chest pain *(cp)*. You also have to include the quadratic term for age and the interaction term between age and maximum heart rate achieved. Before writing any code, review Section 4 of the Summary Report template to see the questions you will be answering about your model.

Run your scripts to get the outputs of your analysis. Then use the outputs to answer the questions in your summary report.

Note: Use the + (plus) button to add new code blocks, if needed.

In [9]:

```
logit_two <- glm(target ~ thalach + age + sex + exang + cp + I(a
ge^2) + age:thalach, data=heart_data, family="binomial")
summary(logit_two)
```

```
Call:
glm(formula = target ~ thalach + age + sex + exang +
cp + I(age^2) +
    age:thalach, family = "binomial", data = heart_d
ata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4225  -0.6167   0.2083   0.6646   2.5398

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.634e+01  1.205e+01  -1.356 0.175117
thalach      1.390e-01  5.701e-02   2.438 0.014760 *
age          2.049e-01  3.112e-01   0.658 0.510325
sex1        -1.709e+00  3.590e-01  -4.762 1.91e-06 *
**
exang1      -9.348e-01  3.586e-01  -2.607 0.009133 *
*
cp1          1.766e+00  4.821e-01   3.663 0.000249 *
**
cp2          1.820e+00  3.844e-01   4.734 2.21e-06 *
**
cp3          1.674e+00  5.764e-01   2.904 0.003684 *
*
I(age^2)     4.921e-04  2.054e-03   0.240 0.810599
thalach:age -2.017e-03  9.999e-04  -2.017 0.043666 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
0.1 ' ' 1

(Dispersion parameter for binomial family taken to b
e 1)

    Null deviance: 417.64  on 302  degrees of freedo
m
Residual deviance: 263.42  on 293  degrees of freedo
m
AIC: 283.42

Number of Fisher Scoring iterations: 5
```

In [10]:

```
# Hosmer-Lemeshow GOF
library(ResourceSelection)

print("Hosmer-Lemeshow Goodness of Fit Test")
hl = hoslem.test(logit_two$y, fitted(logit_two), g=50)
hl

# Wald Test
conf_int_two <- confint.default(logit_two, level=0.95)
round(conf_int_two,4)
```

```
[1] "Hosmer-Lemeshow Goodness of Fit Test"

        Hosmer and Lemeshow goodness of fit (GOF) te
st

data:  logit_two$y, fitted(logit_two)
X-squared = 60.596, df = 48, p-value = 0.1048
```

A matrix: 10 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	-39.9672	7.2803
thalach	0.0273	0.2507
age	-0.4051	0.8148
sex1	-2.4130	-1.0059
exang1	-1.6377	-0.2320
cp1	0.8209	2.7106
cp2	1.0662	2.5732
cp3	0.5442	2.8035
I(age^2)	-0.0035	0.0045
thalach:age	-0.0040	-0.0001

In [11]:

```
# confusion matrix
# Predict target yes or no for the data set using the model
matrix_model_two <- heart_data[c('thalach', 'age', 'sex', 'exang
', 'cp')]

pred_two <- predict(logit_two, newdata=matrix_model_two, type='r
esponse')

depvar_pred_two = as.factor(ifelse(pred_two >= 0.5, '1', '0'))

# This creates the confusion matrix
conf.matrix_two <- table(heart_data$target, depvar_pred_two)[c('
0','1'),c('0','1')]
rownames(conf.matrix_two) <- paste("Actual", rownames(conf.matri
x_two), sep = ": Target=")
colnames(conf.matrix_two) <- paste("Prediction", colnames(conf.m
atrix_two), sep = ": Target=")

# Nicely formatted confusion matrix
print("Confusion Matrix")
format(conf.matrix_two,justify="centre",digit=2)
```

```
[1] "Confusion Matrix"
```

A matrix: 2 × 2 of type chr

	Prediction: Target=0	**Prediction: Target=1**
Actual: Target=0	103	35
Actual: Target=1	27	138

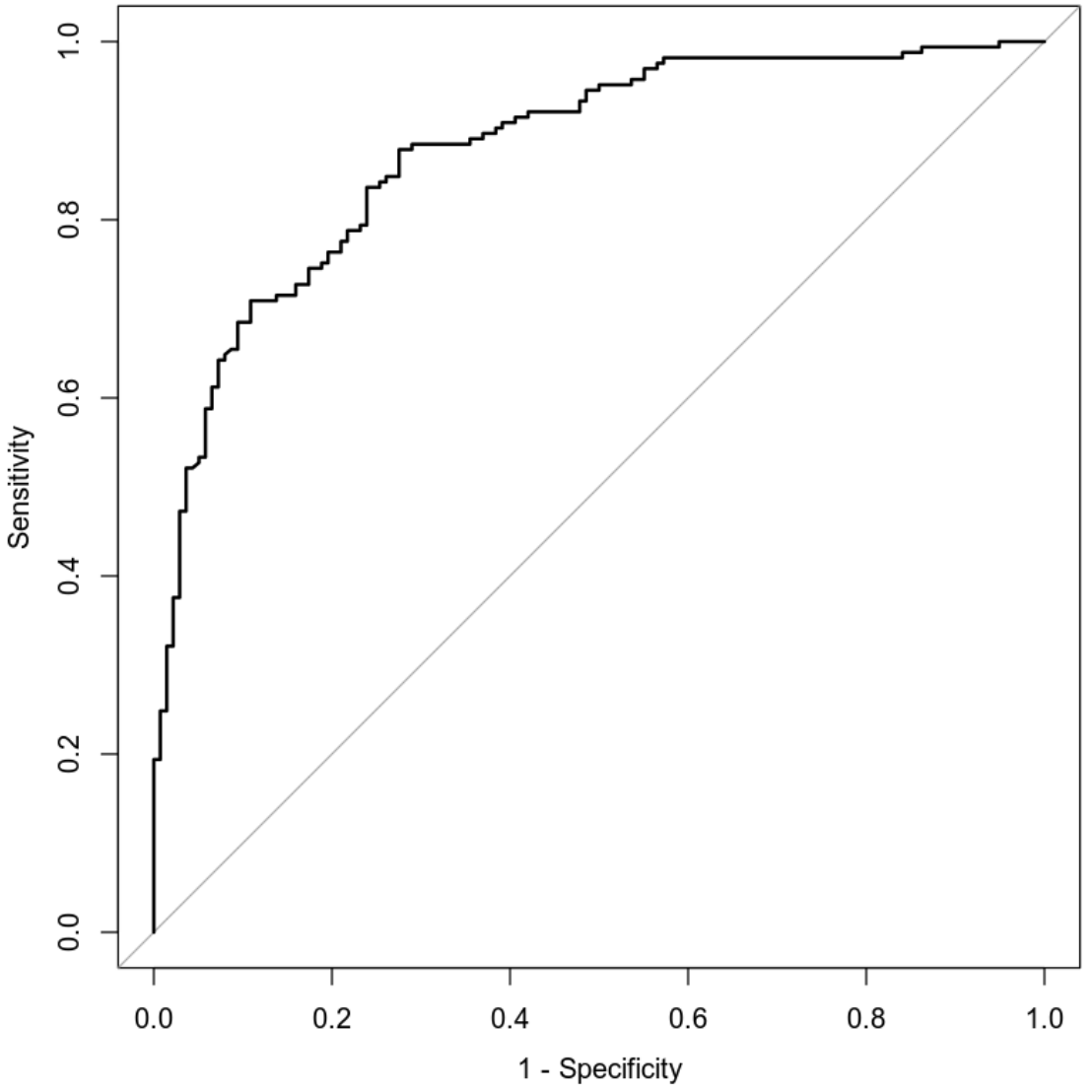
In [12]:

```
library(pROC)

labels2 <- heart_data$target
predictions2 <- logit_two$fitted.values

roc2 <- roc(labels2 ~ predictions2)

print("Area Under the Curve (AUC)")
round(auc(roc2),4)

print("ROC Curve")
# True Positive Rate (Sensitivity) and False Positive Rate (1 - Specificity)
plot(roc2, legacy.axes = TRUE)
```

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

[1] "Area Under the Curve (AUC)"
```

0.8777

```
[1] "ROC Curve"
```



In [20]:

```
print("Prediction: sex = male, Age = 30, rmax heart rate = 145,
exang = yes, chest pain = no")
newdata_two <- data.frame(sex = "1", age=30, thalach=145, exang
= "1", cp = "0")
pred1_two <- predict(logit_two, newdata_two, type='response')
round(pred1_two, 4)


print("Prediction: sex = male, Age = 30, rmax heart rate = 145,
exang = no, chest pain = typical angina")
newdata2_two <- data.frame(sex = "1", age=30, thalach=145, exang
= "1", cp = "1")
pred2_two <- predict(logit_two, newdata2_two, type='response')
round(pred2_two, 4)


odds3 <- ((pred1_two) /(1 - pred1_two))
print("Odds for first prediction")
round(odds1, 4)

odds4 <- ((pred2_two) /(1 - pred2_two))
print("Odds for second prediction")
round(odds2, 4)
```

```
[1] "Prediction: sex = male, Age = 30, rmax heart ra
te = 145, exang = yes, chest pain = no"
```

1: 0.2654

```
[1] "Prediction: sex = male, Age = 30, rmax heart ra
te = 145, exang = no, chest pain = typical angina"
```

1: 0.6787

```
[1] "Odds for first prediction"
```

1: 0.9761

```
[1] "Odds for second prediction"
```

1: 2.6335

Random Forest Classification Model

You have been asked to create a random forest classification model for the presence of heart disease *(target)* using the variables age *(age)*, sex *(sex)*, chest pain type *(cp)*, resting blood pressure *(trestbps)*, cholesterol measurement *(chol)*, resting electrocardiographic measurement *(restecg)*, exercise-induced angina *(exang)*, slope of peak exercise *(slope)*, and number of major vessels *(ca)*. Before writing any code, review Section 5 of the Summary Report template to see the questions you will be answering about your model.

Run your scripts to get the outputs of your regression analysis. Then use the outputs to answer the questions in your summary report.

Note: Use the + (plus) button to add new code blocks, if needed.

In [13]:

```
set.seed(511038)
# splitting for training and testing
samp.size = floor(0.80*nrow(heart_data))
print("Number of rows for the training set")
train_ind = sample(seq_len(nrow(heart_data)), size = samp.size)
train.data = heart_data[train_ind,]
nrow(train.data)
print("Number of rows for the testing set")
test.data = heart_data[-train_ind,]
nrow(test.data)
```

```
[1] "Number of rows for the training set"
```

242

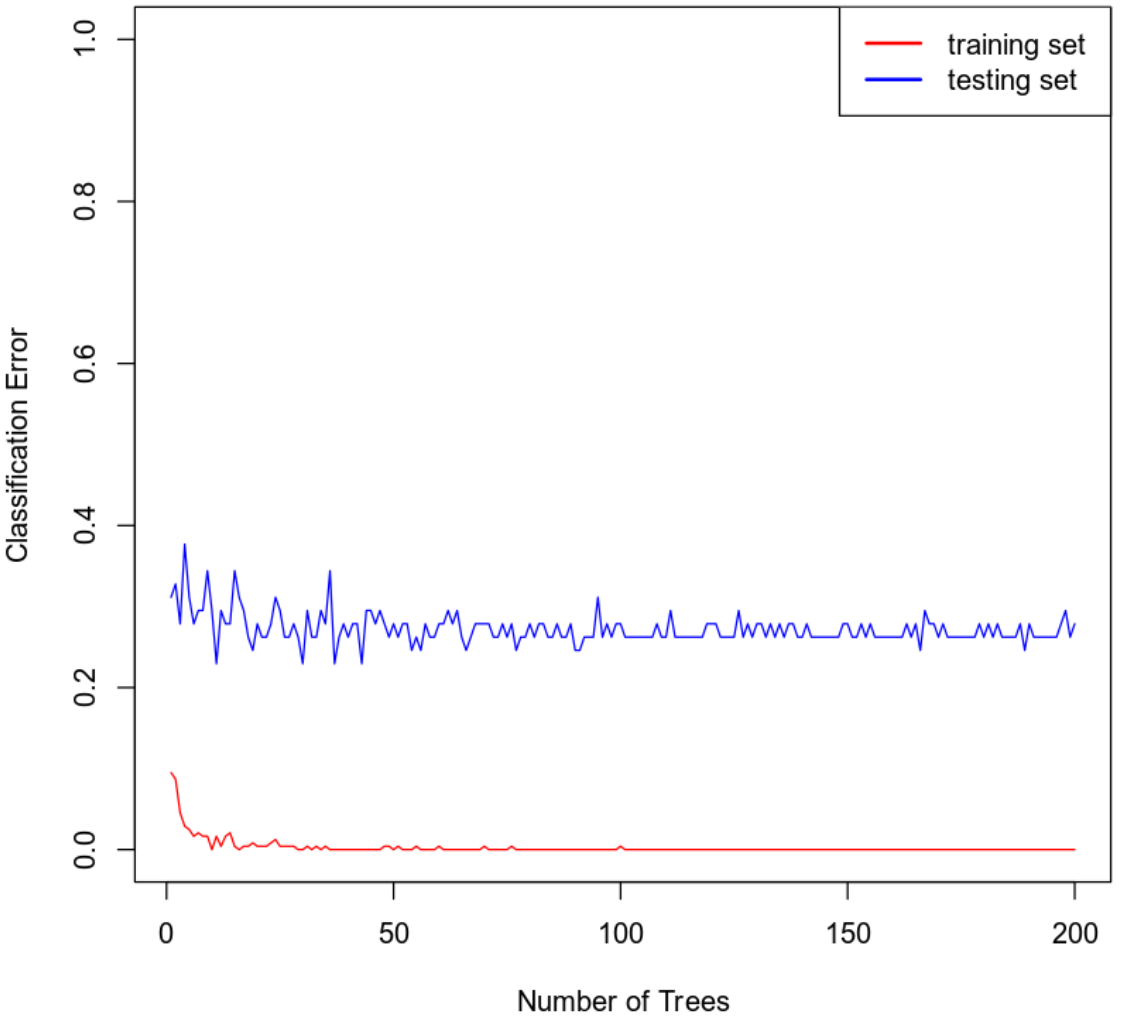
```
[1] "Number of rows for the testing set"
```

61

In [14]:

```
set.seed(511038)
library(randomForest)
train = c()
test = c()
trees = c()
for(i in seq(from=1, to=200, by=1)) {
    #print(i)

    trees <- c(trees, i)

    model_rf3 <- randomForest(target ~ age+sex+cp+trestbps+chol+
restecg+exang+slope+ca, data=train.data, ntree=i)

    train.data.predict <- predict(model_rf3, train.data, type =
"class")
    conf.matrix1 <- table(train.data$target, train.data.predict)
    train_error = 1-(sum(diag(conf.matrix1)))/sum(conf.matrix1)
    train <- c(train, train_error)

    test.data.predict <- predict(model_rf3, test.data, type = "c
lass")
    conf.matrix2 <- table(test.data$target, test.data.predict)
    test_error = 1-(sum(diag(conf.matrix2)))/sum(conf.matrix2)
    test <- c(test, test_error)
}

plot(trees, train,type = "l",ylim=c(0,1.0),col = "red", xlab = "
Number of Trees", ylab = "Classification Error")
lines(test, type = "l", col = "blue")
legend('topright',legend = c('training set','testing set'), col
= c("red","blue"), lwd = 2 )
```

```
randomForest 4.6-14
Type rfNews() to see new features/changes/bug fixes.
```



In [15]:

```
set.seed(511038)
library(randomForest)
model_ranfor <- randomForest(target ~ age+sex+cp+trestbps+chol+r
estecg+exang+slope+ca, data=train.data, ntree=20)
# confusion Matrix

print('Confusion Matrix for Training set based on 20 trees')
train.data.predict <- predict(model_ranfor, train.data, type = "
class")

# Construct the confusion matrix
conf.matrix <- table(train.data$target, train.data.predict)
rownames(conf.matrix) <- paste("Actual", rownames(conf.matrix),
sep = ": ")
colnames(conf.matrix) <- paste("Prediction", colnames(conf.matri
x), sep = ": ")

# Print nicely formatted confusion matrix
format(conf.matrix,justify="centre",digit=2)

print('Confusion Matrix for Testing set based on 20 trees')
test.data.predict <- predict(model_ranfor, test.data, type = "cl
ass")

# Construct the confusion matrix
conf.matrix_two <- table(test.data$target, test.data.predict)
rownames(conf.matrix_two) <- paste("Actual", rownames(conf.matri
x_two), sep = ": ")
colnames(conf.matrix_two) <- paste("Prediction", colnames(conf.m
atrix_two), sep = ": ")

# Print nicely formatted confusion matrix
format(conf.matrix_two,justify="centre",digit=2)
```

```
[1] "Confusion Matrix for Training set based on 20 t
rees"
```

A matrix: 2 × 2 of type chr

	Prediction: 0	**Prediction: 1**
Actual: 0	111	1
Actual: 1	0	130

```
[1] "Confusion Matrix for Testing set based on 20 tr
ees"
```

A matrix: 2 × 2 of type chr

	Prediction: 0	**Prediction: 1**
Actual: 0	18	8
Actual: 1	7	28

In [16]:

```
print('Training set Accuracy:')
print((130+111)/(130+111+0+1))
print('_____________')

print('Training set Precision:')
print((130)/(130+1))
print('_____________')

print('Training set Recall:')
print((130)/(130+0))
print('_____________________________________')

print('Testing set Accuracy:')
print((28+17)/(28+17+7+9))
print('_____________')

print('Testing set Precision:')
print((28)/(28+9))
print('_____________')

print('Testing set Recall:')
print((28)/(28+7))
```

```
[1] "Training set Accuracy:"
[1] 0.9958678
[1] "_____________"
[1] "Training set Precision:"
[1] 0.9923664
[1] "_____________"
[1] "Training set Recall:"
[1] 1
[1] "_____________________________________"
[1] "Testing set Accuracy:"
[1] 0.7377049
[1] "_____________"
[1] "Testing set Precision:"
[1] 0.7567568
[1] "_____________"
[1] "Testing set Recall:"
[1] 0.8
```

Random Forest Regression Model

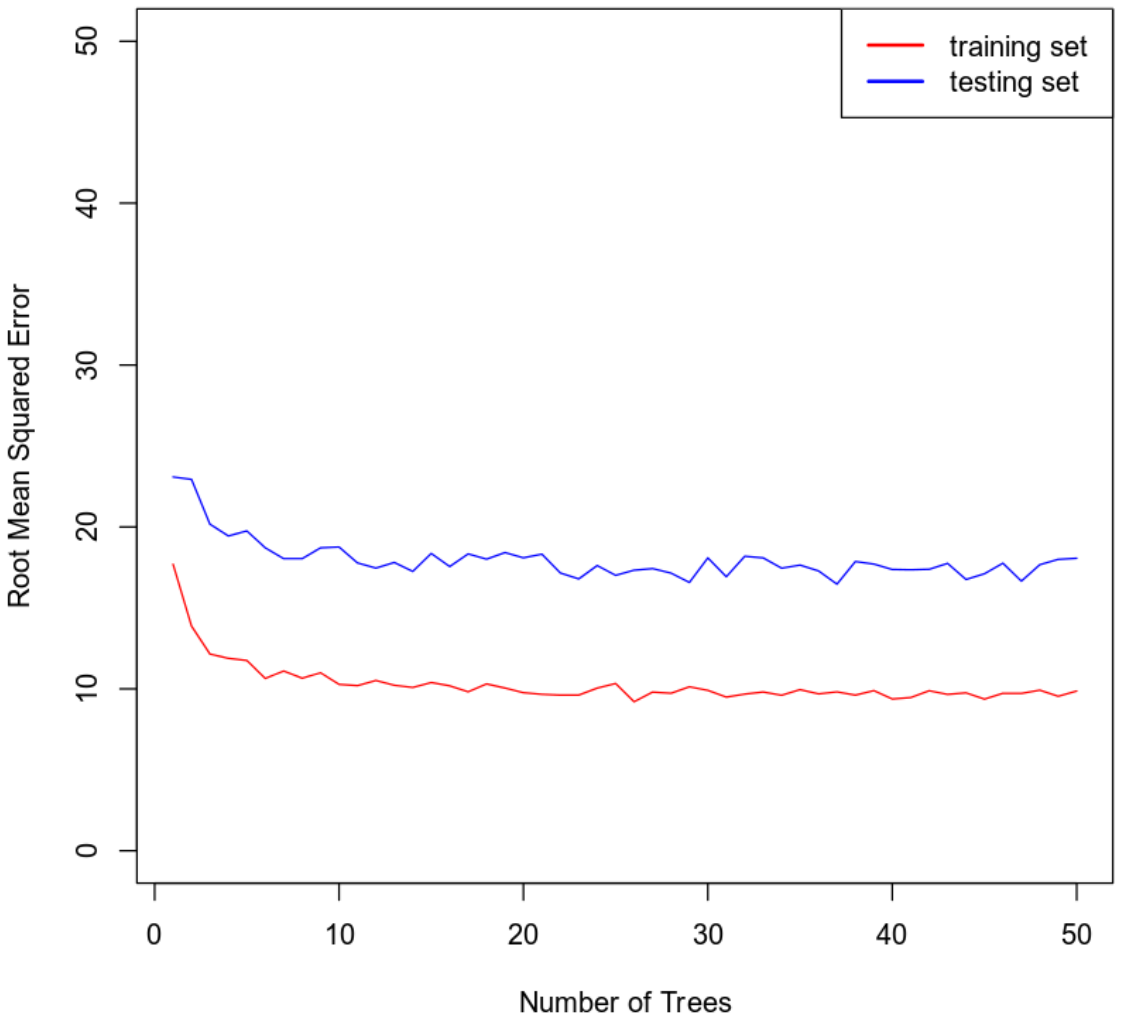
You have been asked to create a random forest regression model for maximum heart rate achieved using the variables age *(age)*, sex *(sex)*, chest pain type *(cp)*, resting blood pressure *(trestbps)*, cholesterol measurement *(chol)*, resting electrocardiographic measurement *(restecg)*, exercise-induced angina *(exang)*, slope of peak exercise *(slope)*, and number of major vessels *(ca)*. Before writing any code, review Section 6 of the Summary Report template to see the questions you will be answering about your model.

Run your scripts to get the outputs of your analysis. Then use the outputs to answer the questions in your summary report.

Note: Use the + (plus) button to add new code blocks, if needed.

In [17]:

```
set.seed(511038)
library(randomForest)

# Root mean squared error
RMSE = function(pred, obs) {
    return(sqrt( sum( (pred - obs)^2 )/length(pred) ) )
}


# checking
#=============================================================
======
train = c()
test = c()
trees = c()

for(i in seq(from=1, to=50, by=1)) {
    trees <- c(trees, i)
    model_regfor <- randomForest(thalach ~ age+sex+cp+trestbps+c
hol+restecg+exang+slope+ca, data=train.data, ntree = i)

    pred <- predict(model_regfor, newdata=train.data, type='resp
onse')
    rmse_train <-  RMSE(pred, train.data$thalach)
    train <- c(train, rmse_train)

    pred <- predict(model_regfor, newdata=test.data, type='respo
nse')
     rmse_test <-  RMSE(pred, test.data$thalach)
    test <- c(test, rmse_test)
}

plot(trees, train,type = "l",ylim=c(0,50),col = "red", xlab = "N
umber of Trees", ylab = "Root Mean Squared Error")
lines(test, type = "l", col = "blue")
legend('topright',legend = c('training set','testing set'), col
= c("red","blue"), lwd = 2 )
```


training set
testing set
Root Mean Squared Error
0
10
20
30
40
50
Number of Trees

In [18]:

```
set.seed(511038)
library(randomForest)

model_regfor2 <- randomForest(thalach ~ age+sex+cp+trestbps+chol
+restecg+exang+slope+ca, data=train.data, ntree = 15)

# RMSE
RMSE = function(pred, obs) {
    return(sqrt( sum( (pred - obs)^2 )/length(pred) ) )
}

print('Root Mean Squared Error: TRAINING set based on random for
est model built using 15 trees')
pred <- predict(model_regfor2, newdata=train.data, type='respons
e')
RMSE(pred, train.data$thalach)


print('Root Mean Squared Error: TESTING set based on random fore
st model built using 15 trees')
pred2 <- predict(model_regfor2, newdata=test.data, type='respons
e')
RMSE(pred2, test.data$thalach)
```

```
[1] "Root Mean Squared Error: TRAINING set based on
random forest model built using 15 trees"
```

10.3173274982241

```
[1] "Root Mean Squared Error: TESTING set based on r
andom forest model built using 15 trees"
```

18.0448389248858

End of Project Two Jupyter Notebook

The HTML output can be downloaded by clicking **File**, then **Download as**, then **HTML**. Be sure to answer all of the questions in the Summary Report template for Project Two, and to include your completed Jupyter Notebook scripts as part of your submission.