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 [2]:
```
install.packages("ResourceSelection")
install.packages("pROC")
install.packages("rpart.plot")
```

```
Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.
4'
(as 'lib' is unspecified)
Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.
4'
(as 'lib' is unspecified)
Installing package into '/home/codio/R/x86_64-pc-linux-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	oldpeak	slope	ca
<int>	<fct>	<fct>	<int>	<int>	<fct>	<fct>	<int>	<fct>	<dbl>	<fct>	<fct>
62	1	2	130	231	0	1	146	0	1.8	1	3
58	0	0	130	197	0	1	131	0	0.6	1	0
60	0	3	150	240	0	1	171	0	0.9	2	0
63	1	0	140	187	0	0	144	1	4.0	2	2
62	1	0	120	267	0	1	99	1	1.8	1	2
63	0	2	135	252	0	0	172	0	0.0	2	0
43	1	0	150	247	0	1	171	0	1.5	2	0
42	1	2	120	240	1	1	194	0	0.8	0	0
59	1	2	126	218	1	1	134	0	2.2	1	1
48	1	0	124	274	0	0	166	0	0.5	1	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 [22]:
```
logit1 <- glm(target ~ age + trestbps + thalach, data = heart_data, fa
mily = "binomial")

summary(logit1)
```

```
Call:
glm(formula = target ~ age + trestbps + thalach, family = "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 be 1)

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

Number of Fisher Scoring iterations: 3
```

In [23]:
```
library(ResourceSelection)

print("Hosmer-Lemeshow Goodness of Fit Test")
hl = hoslem.test(logit1$y, fitted(logit1), g=50)
hl
```

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

        Hosmer and Lemeshow goodness of fit (GOF) test

data:  logit1$y, fitted(logit1)
X-squared = 41.978, df = 48, p-value = 0.7168
```

In [30]:
```
# predict heart disease or no heart disease for the dataset using the
model
default_model_data <- heart_data[c('age', 'trestbps', 'thalach')]
pred <- predict(logit1, newdata=default_model_data, type='response')

# if the predicted probability of heart disease is >=0.50 then predict
heart disease (default='1'), otherwise predict no heart
# disease (default='0')
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 =
": default=")
colnames(conf.matrix) <- paste("Prediction", colnames(conf.matrix), se
p = ": default=")


# print 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: default=0	Prediction: default=1
Actual: default=0	83	55
Actual: default=1	38	127

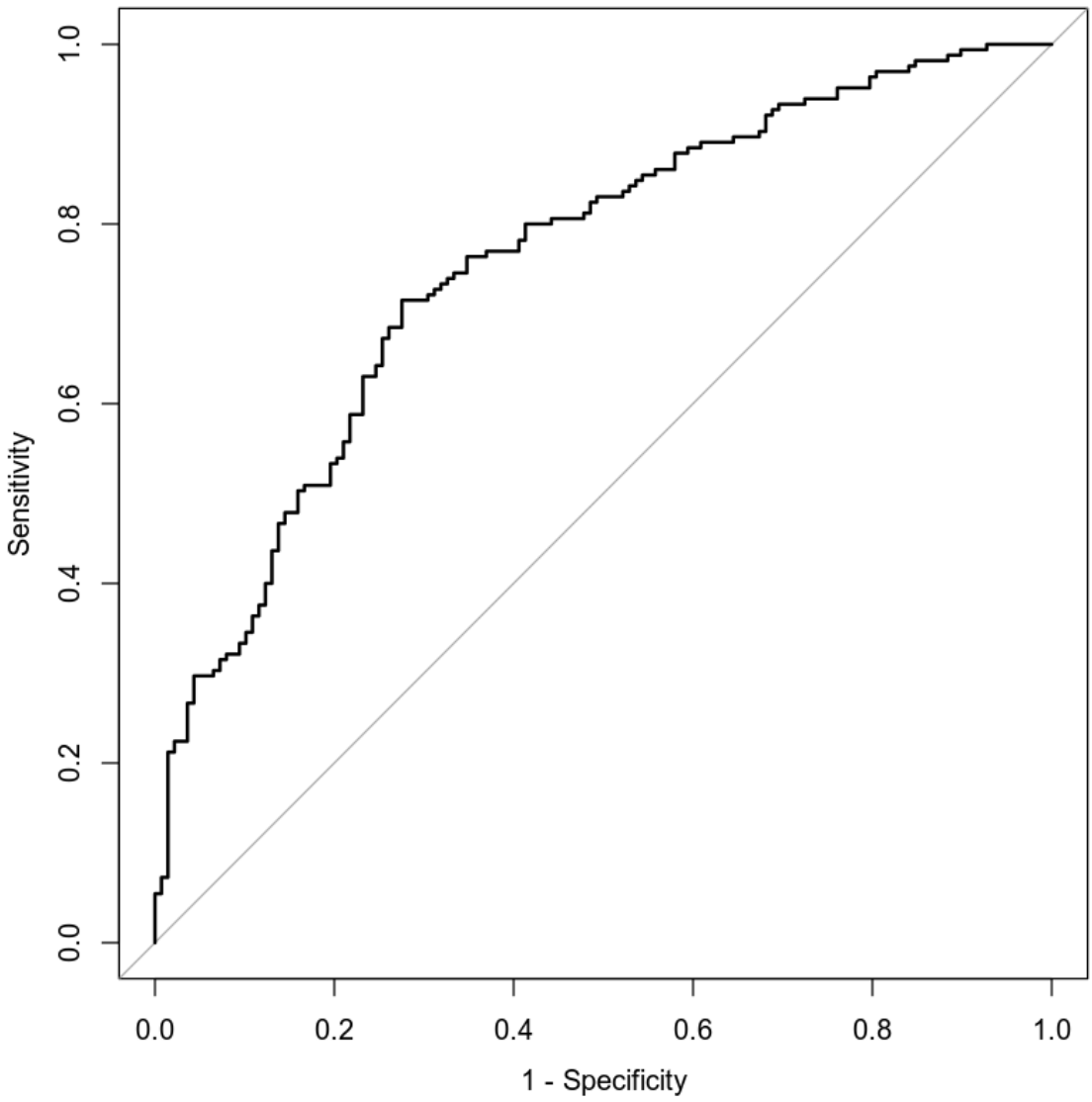
In [27]:
```
library(pROC)

labels <- heart_data$target
predictions = logit1$fitted.values

roc <- roc(labels ~ predictions)

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

# Print ROC Curve
print("ROC Curve")

# True Positive Rate (Sensitivity) and False Positive Rate (1 - Specif
icity)
plot(roc, legacy.axes = TRUE)
```

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

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

```
0.7575
```

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



In [28]:
```
# Prediction of heart disease if age=50, resting blood pressure is 12
2, and max heart rate is 140
print("Prediction: age=50, trestbps=122, thalach=140")
newdata1 <- data.frame(age=50, trestbps=122, thalach=140)
round(predict(logit1, newdata1, type='response'), 4)

# Prediction of heart disease if age=50, resting blood pressure is 14
0, and max heart rate is 170
print("Prediction: age=50, trestbps=140, thalach=170")
newdata1 <- data.frame(age=50, trestbps=140, thalach=170)
round(predict(logit1, newdata1, type='response'), 4)
```

```
[1] "Prediction: age=50, trestbps=122, thalach=140"
```

1: 0.4939

```
[1] "Prediction: age=50, trestbps=140, thalach=170"
```

1: 0.7248

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 [4]:
```
logit2 <- glm(target ~ thalach + age + sex + exang + cp + I(age^2) + a
ge:thalach, data = heart_data, family = "binomial")

summary(logit2)
```

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

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 be 1)

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

Number of Fisher Scoring iterations: 5
```

In [7]:
```
library(ResourceSelection)

print("Hosmer-Lemeshow Goodness of Fit Test")
hl = hoslem.test(logit2$y, fitted(logit2), g=50)
hl

# predict heart disease or no heart disease for the dataset using the
model
default_model_data2 <- heart_data[c('thalach', 'age', 'sex', 'exang',
'cp')]
pred2 <- predict(logit2, newdata=default_model_data2, type='response')

# if the predicted probability of heart disease is >=0.50 then predict
heart disease (default='1'), otherwise predict no heart
# disease (default='0')
depvar_pred2 = as.factor(ifelse(pred2 >= 0.5, '1', '0'))
# this creates the confusion matrix
conf.matrix <- table(heart_data$target, depvar_pred2)[c('0','1'),c
('0','1')]
rownames(conf.matrix) <- paste("Actual", rownames(conf.matrix), sep =
": default=")
colnames(conf.matrix) <- paste("Prediction", colnames(conf.matrix), se
p = ": default=")

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

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

        Hosmer and Lemeshow goodness of fit (GOF) test

data:  logit2$y, fitted(logit2)
X-squared = 60.596, df = 48, p-value = 0.1048

[1] "Confusion Matrix"
```

A matrix: 2 × 2 of type chr

	Prediction: default=0	Prediction: default=1
Actual: default=0	103	35
Actual: default=1	27	138

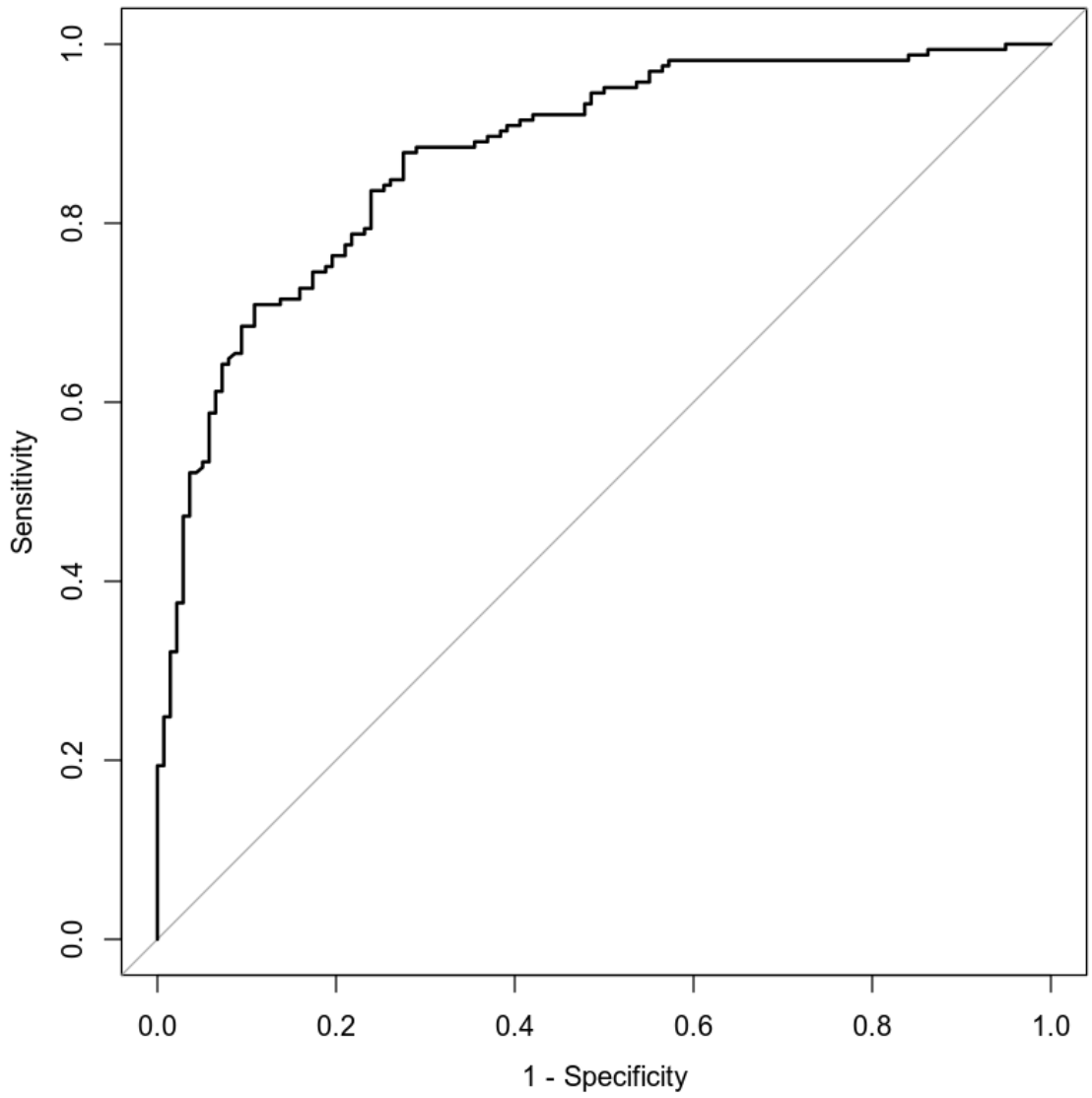
In [13]:
```
library(pROC)

labels <- heart_data$target
predictions = logit2$fitted.values

roc <- roc(labels ~ predictions)

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

# Print ROC Curve
print("ROC Curve")

# True Positive Rate (Sensitivity) and False Positive Rate (1 - Specif
icity)
plot(roc, 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 [14]:
```
# Prediction of heart disease if age=30, sex=male('1'), max heart rate
=145, exang='1' and cp='0'
print("Prediction: age=30, sex='1', thalach=145, exang='1', cp='0'")
newdata2 <- data.frame(age=30, sex='1', thalach=145, exang='1', cp
='0')
round(predict(logit2, newdata2, type='response'), 4)

# Prediction of heart disease if age=30, sex=male('1'), max heart rate
=145, exang='0' and cp='1'
print("Prediction: age=30, sex='1', thalach=145, exang='0', cp='1'")
newdata2 <- data.frame(age=30, sex='1', thalach=145, exang='0', cp
='1')
round(predict(logit2, newdata2, type='response'), 4)
```

```
[1] "Prediction: age=30, sex='1', thalach=145, exang='1', cp='0'"
```

1: 0.2654

```
[1] "Prediction: age=30, sex='1', thalach=145, exang='0', cp='1'"
```

1: 0.8432

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 [3]:
```
set.seed(511038)

# Partition the data set into training and testing data
samp.size = floor(0.80*nrow(heart_data))

# Training set
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)

# Testing set
print("Number of rows for the testing set")
test.data = heart_data[-train_ind,]
nrow(test.data)
```

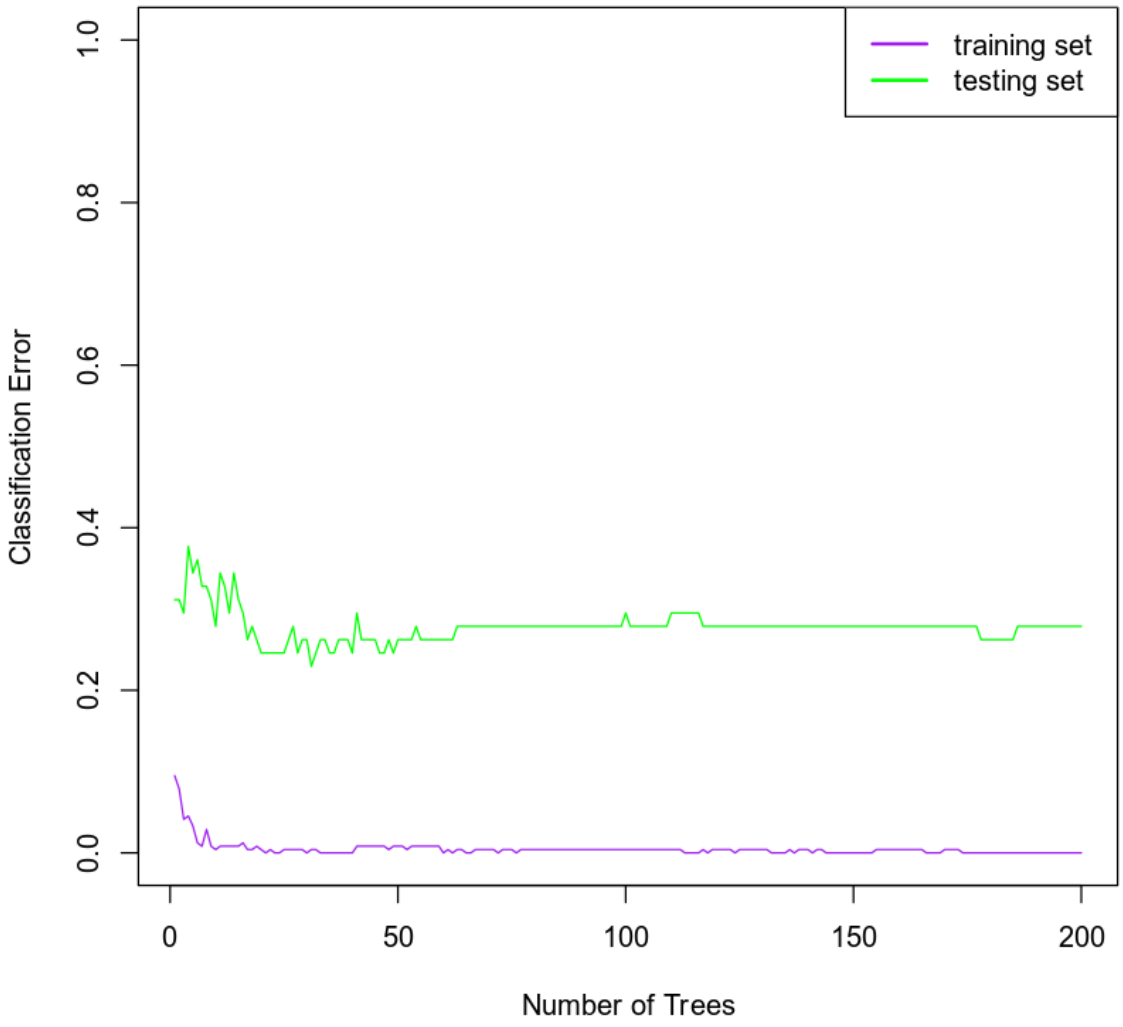
```
[1] "Number of rows for the training set"
```

```
242
```

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

```
61
```

In [6]:
```
library(randomForest)

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

for(i in seq(from=1, to=200, by=1)) {
    #print(i)

    trees <- c(trees, i)
    set.seed(511038)
    model_rf1 <- randomForest(target ~ age+sex+cp+trestbps+chol+restec
g+exang+slope+ca, data=train.data, ntree = i)

    train.data.predict <- predict(model_rf1, train.data, type = "clas
s")
    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_rf1, test.data, type = "class")
    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),col = "purple", xlab = "Numbe
r of Trees", ylab = "Classification Error")
lines(test, type = "l", col = "green")
legend('topright',legend = c('training set','testing set'), col = c("p
urple","green"), lwd = 2 )
```


training set
testing set
Classification Error
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
200
Number of Trees

In [8]:
```
set.seed(511038)
library(randomForest)
model_rf1 <- randomForest(target ~ age+sex+cp+trestbps+chol+restecg+ex
ang+slope+ca, data=train.data, ntree = 20)

# Confusion Matrix
print("============================================================
==========================================================")
print('Confusion Matrix: TRAINING set')
train.data.predict <- predict(model_rf1, train.data, type = "class")

# construct the confusion matrix
conf.matrix1 <- table(train.data$target, train.data.predict)[,c
('0','1')]
rownames(conf.matrix1) <- paste("Actual", rownames(conf.matrix1), sep
= ": ")
colnames(conf.matrix1) <- paste("Prediction", colnames(conf.matrix1),
sep = ": ")

# print nicely formatted confusion matrix
format(conf.matrix1,justify="centre",digit=2)


print("============================================================
==========================================================")
print('Confusion Matrix: TESTING set')
test.data.predict <- predict(model_rf1, test.data, type = "class")

# construct the confusion matrix
conf.matrix2 <- table(test.data$target, test.data.predict)[,c
('0','1')]
rownames(conf.matrix2) <- paste("Actual", rownames(conf.matrix2), sep
= ": ")
colnames(conf.matrix2) <- paste("Prediction", colnames(conf.matrix2),
sep = ": ")

# print nicely formatted confusion matrix
format(conf.matrix2,justify="centre",digit=2)
```

```
[1] "============================================================
============================================================"
[1] "Confusion Matrix: TRAINING set"
```

A matrix: 2 × 2 of type chr

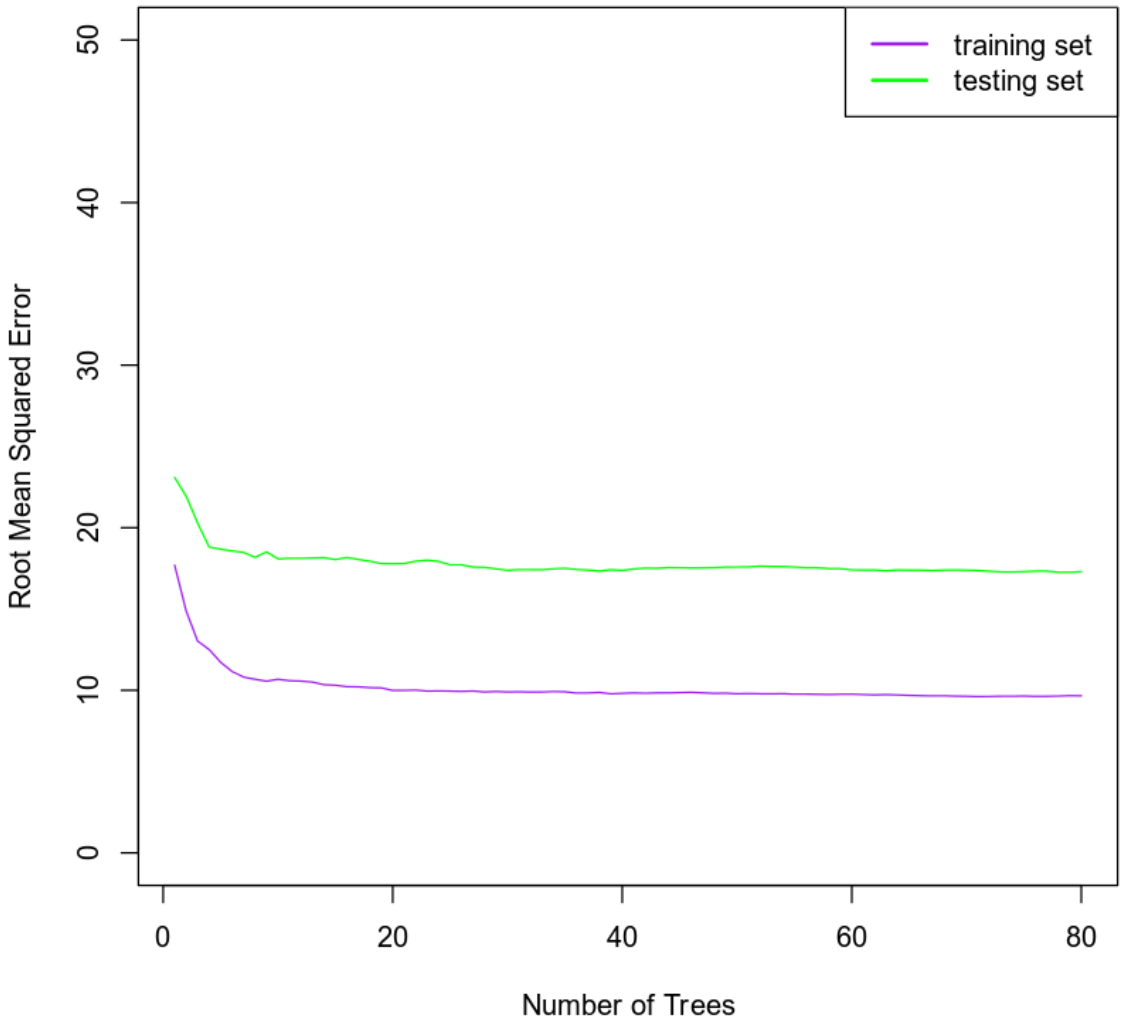
	Prediction: 0	**Prediction: 1**
Actual: 0	111	1
Actual: 1	0	130

```
[1] "============================================================
============================================================"
[1] "Confusion Matrix: TESTING set"
```

A matrix: 2 × 2 of type chr

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

In [10]:
```
# 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=80, by=1)) {
    set.seed(511038)
    trees <- c(trees, i)
    model_rf2 <- randomForest(thalach ~ age+sex+cp+trestbps+chol+reste
cg+exang+slope+ca, data=train.data, ntree = i)

    pred <- predict(model_rf2, newdata=train.data, type='response')
    rmse_train <-  RMSE(pred, train.data$thalach)
    rmse_train
    train <- c(train, rmse_train)

    pred <- predict(model_rf2, newdata=test.data, type='response')
    rmse_test <-  RMSE(pred, test.data$thalach)
    test <- c(test, rmse_test)
}

plot(trees, train,type = "l",col = "purple", ylim=c(0,50), xlab = "Num
ber of Trees", ylab = "Root Mean Squared Error")
lines(test, type = "l", col = "green")
legend('topright',legend = c('training set','testing set'), col = c("p
urple","green"), lwd = 2 )
```


training set
testing set
Root Mean Squared Error
0
10
20
30
40
50
0
20
40
60
80
Number of Trees

In [16]:
```
set.seed(511038)
model_rf2 <- randomForest(thalach ~ age+sex+cp+trestbps+chol+restecg+e
xang+slope+ca, data=train.data, ntree = 30)

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

print('Root Mean Squared Error: TRAINING set')
pred <- predict(model_rf2, newdata=train.data, type='response')
round(RMSE(pred, train.data$thalach),4)



print('Root Mean Squared Error: TESTING set')
pred <- predict(model_rf2, newdata=test.data, type='response')
round(RMSE(pred, test.data$thalach),4)
```

```
[1] "Root Mean Squared Error: TRAINING set"
```

9.9028

```
[1] "Root Mean Squared Error: TESTING set"
```

17.387

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 [ ]:
```
# 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=80, by=1)) {
    set.seed(511038)
    trees <- c(trees, i)
    model_rf2 <- randomForest(thalach ~ age+sex+cp+trestbps+chol+reste
cg+exang+slope+ca, data=train.data, ntree = i)

    pred <- predict(model_rf2, newdata=train.data, type='response')
    rmse_train <-  RMSE(pred, train.data$thalach)
    rmse_train
    train <- c(train, rmse_train)

    pred <- predict(model_rf2, newdata=test.data, type='response')
    rmse_test <-  RMSE(pred, test.data$thalach)
    test <- c(test, rmse_test)
}

plot(trees, train,type = "l",col = "purple", ylim=c(0,50), xlab = "Num
ber of Trees", ylab = "Root Mean Squared Error")
lines(test, type = "l", col = "green")
legend('topright',legend = c('training set','testing set'), col = c("p
urple","green"), lwd = 2 )
```

In [ ]:
```
set.seed(511038)
model_rf2 <- randomForest(thalach ~ age+sex+cp+trestbps+chol+restecg+e
xang+slope+ca, data=train.data, ntree = 30)

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

print('Root Mean Squared Error: TRAINING set')
pred <- predict(model_rf2, newdata=train.data, type='response')
round(RMSE(pred, train.data$thalach),4)



print('Root Mean Squared Error: TESTING set')
pred <- predict(model_rf2, newdata=test.data, type='response')
round(RMSE(pred, test.data$thalach),4)
```

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.