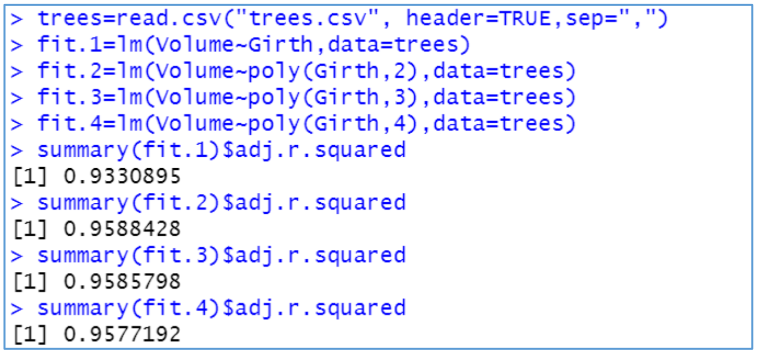
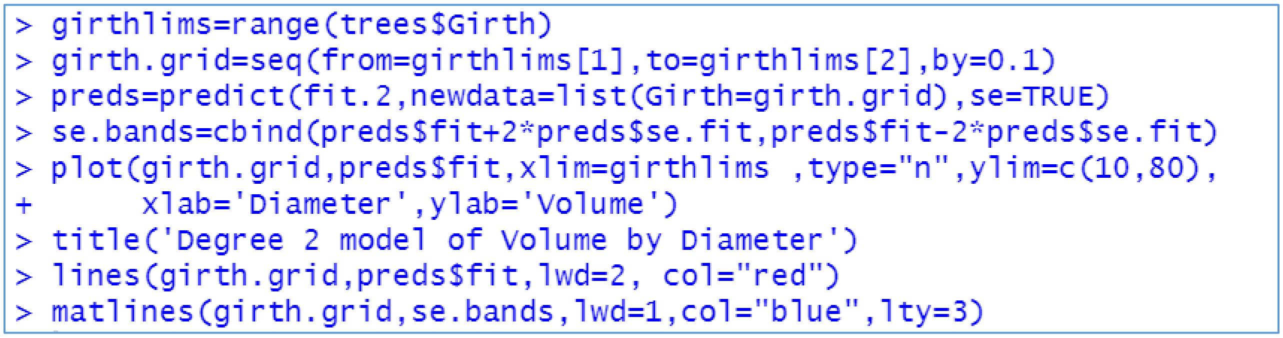
Problem 1: Investigation of the Diameter, Height and Volume for Black Cherry Trees

(1) Four polynomial models (deg=1,2,3,4) were fitted with adjust R-square as below:

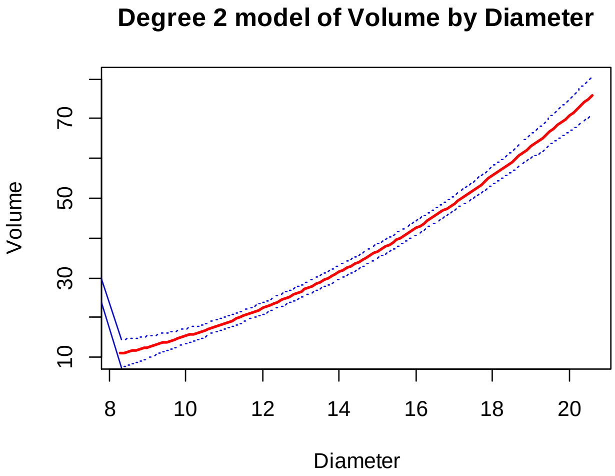


```
> trees=read.csv("trees.csv", header=TRUE,sep=",")
> fit.1=lm(Volume~Girth,data=trees)
> fit.2=lm(Volume~poly(Girth,2),data=trees)
> fit.3=lm(Volume~poly(Girth,3),data=trees)
> fit.4=lm(Volume~poly(Girth,4),data=trees)
> summary(fit.1)$adj.r.squared
[1] 0.9330895
> summary(fit.2)$adj.r.squared
[1] 0.9588428
> summary(fit.3)$adj.r.squared
[1] 0.9585798
> summary(fit.4)$adj.r.squared
[1] 0.9577192
```

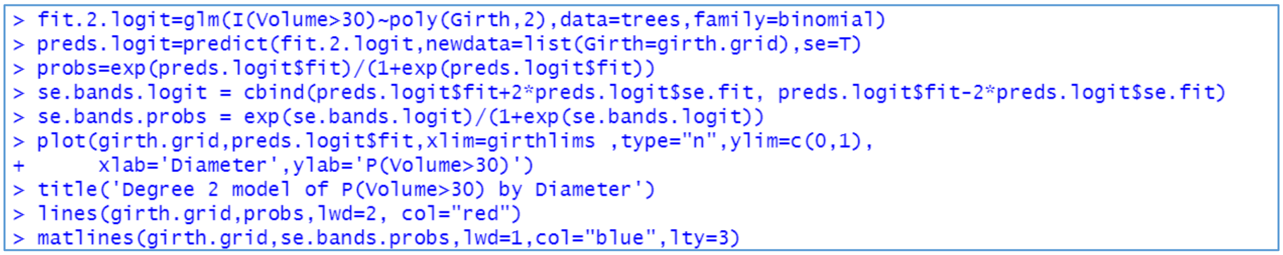
Model with degree 2 has the largest adjust R-squared. Required code and plot is as below:



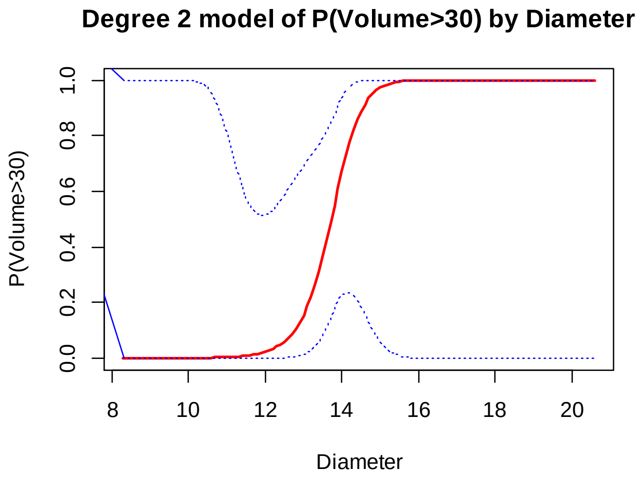
```
> girthlims=range(trees$Girth)
> girth.grid=seq(from=girthlims[1],to=girthlims[2],by=0.1)
> preds=predict(fit.2,newdata=list(Girth=girth.grid),se=TRUE)
> se.bands=cbind(preds$fit+2*preds$se.fit,preds$fit-2*preds$se.fit)
> plot(girth.grid,preds$fit,xlim=girthlims ,type="n",ylim=c(10,80),
+      xlab='Diameter',ylab='Volume')
> title('Degree 2 model of Volume by Diameter')
> lines(girth.grid,preds$fit,lwd=2, col="red")
> matlines(girth.grid,se.bands,lwd=1,col="blue",lty=3)
```



(2) Code and Plot of logistic regression model with deg=2:

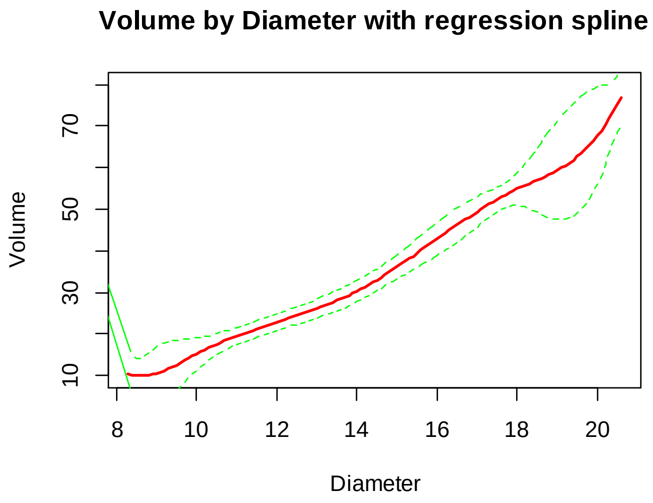


```
> fit.2.logit=glm(I(Volume>30)~poly(Girth,2),data=trees,family=binomial)
> preds.logit=predict(fit.2.logit,newdata=list(Girth=girth.grid),se=T)
> probs=exp(preds.logit$fit)/(1+exp(preds.logit$fit))
> se.bands.logit = cbind(preds.logit$fit+2*preds.logit$se.fit, preds.logit$fit-2*preds.logit$se.fit)
> se.bands.probs = exp(se.bands.logit)/(1+exp(se.bands.logit))
> plot(girth.grid,preds.logit$fit,xlim=girthlims ,type="n",ylim=c(0,1),
+      xlab='Diameter',ylab='P(Volume>30)')
> title('Degree 2 model of P(Volume>30) by Diameter')
> lines(girth.grid,probs,lwd=2, col="red")
> matlines(girth.grid,se.bands.probs,lwd=1,col="blue",lty=3)
```

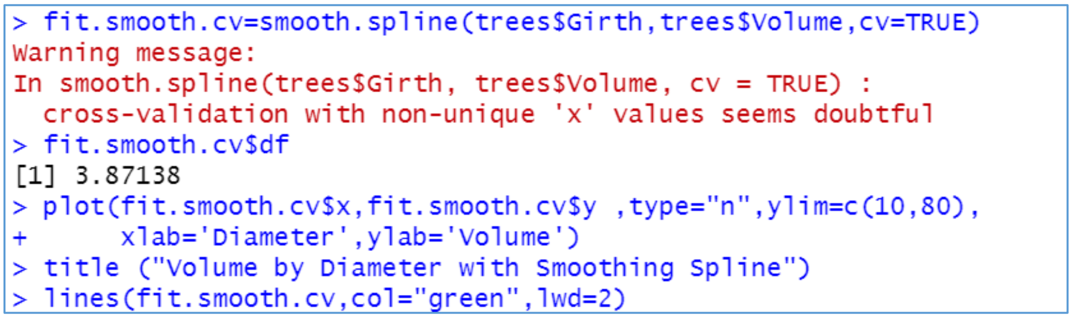

Degree 2 model of P(Volume>30) by Diameter

(3) Spline with deg=2 at knots 10,14,18:

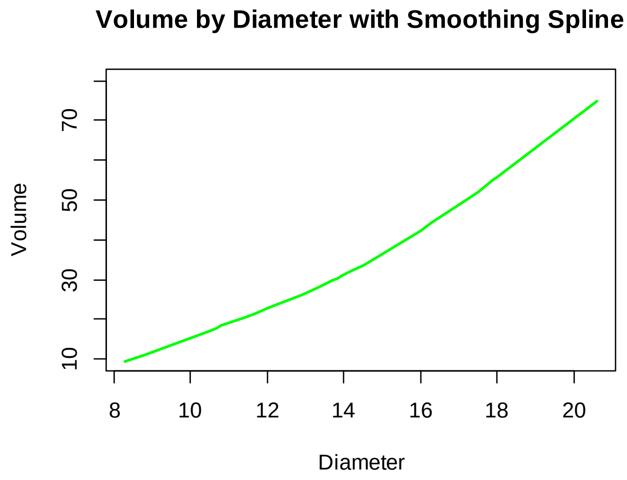
```
library(splines)
fit.spline=lm(Volume~bs(Girth,df=2,knots=c(10,14,18)),data=trees)
preds.spline=predict(fit.spline,newdata=list(Girth=girth.grid),se=T)
plot(girth.grid,preds$fit,xlim=girthlims ,type="n",ylim=c(10,80),
     xlab='Diameter',ylab='Volume')
title('Volume by Diameter with regression spline')
lines(girth.grid,preds.spline$fit,lwd=2,col="red")
lines(girth.grid,preds.spline$fit+2*preds.spline$se ,lty="dashed",col="green")
lines(girth.grid,preds.spline$fit-2*preds.spline$se ,lty="dashed",col="green")
```


Volume by Diameter with regression spline

(4) Using smoothing spline with Cross-Validation, the degree of freedom used was 3.87138.

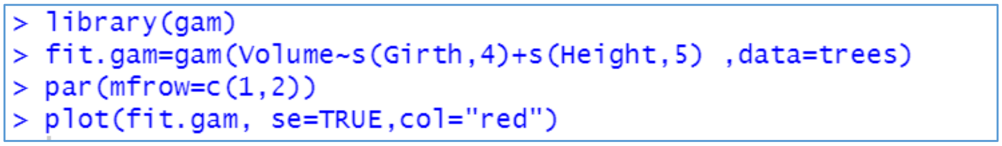


```
> fit.smooth.cv=smooth.spline(trees$Girth,trees$Volume,cv=TRUE)
Warning message:
In smooth.spline(trees$Girth, trees$Volume, cv = TRUE) :
  cross-validation with non-unique 'x' values seems doubtful
> fit.smooth.cv$df
[1] 3.87138
> plot(fit.smooth.cv$x,fit.smooth.cv$y ,type="n",ylim=c(10,80),
+      xlab='Diameter',ylab='Volume')
> title ("Volume by Diameter with Smoothing Spline")
> lines(fit.smooth.cv,col="green",lwd=2)
```

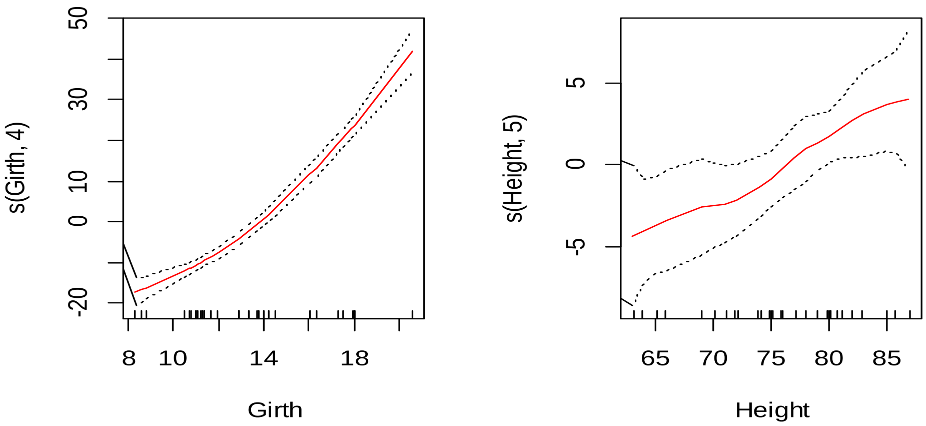


(5) Predict Volume by a GAM with Girth (df=4) and Height (df=5).

Code and function plots:

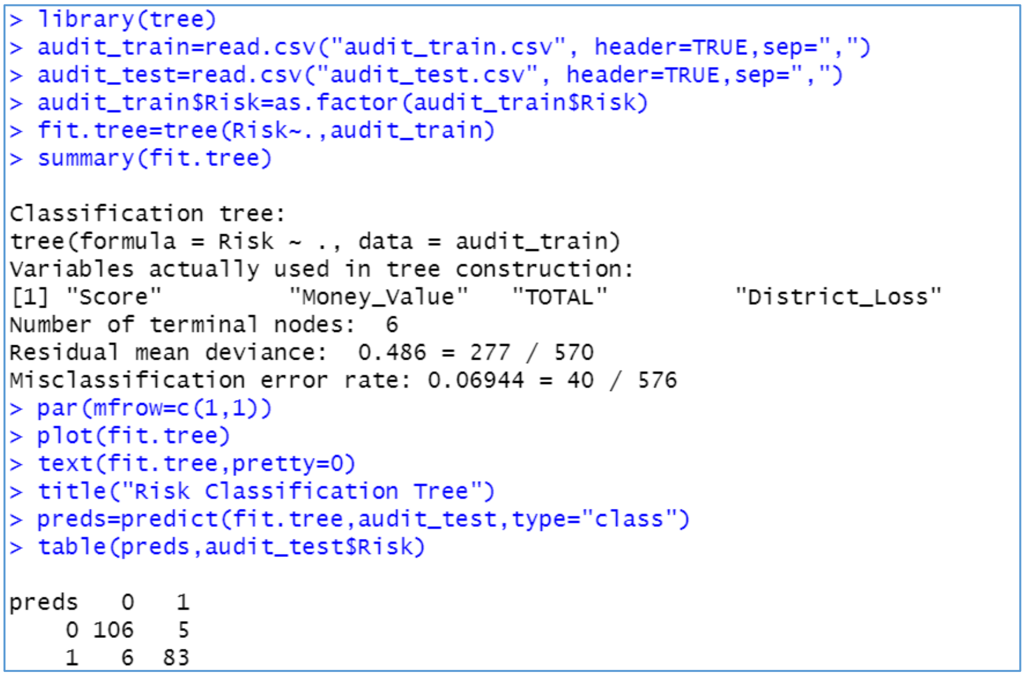


```
> library(gam)
> fit.gam=gam(Volume~s(Girth,4)+s(Height,5) ,data=trees)
> par(mfrow=c(1,2))
> plot(fit.gam, se=TRUE,col="red")
```

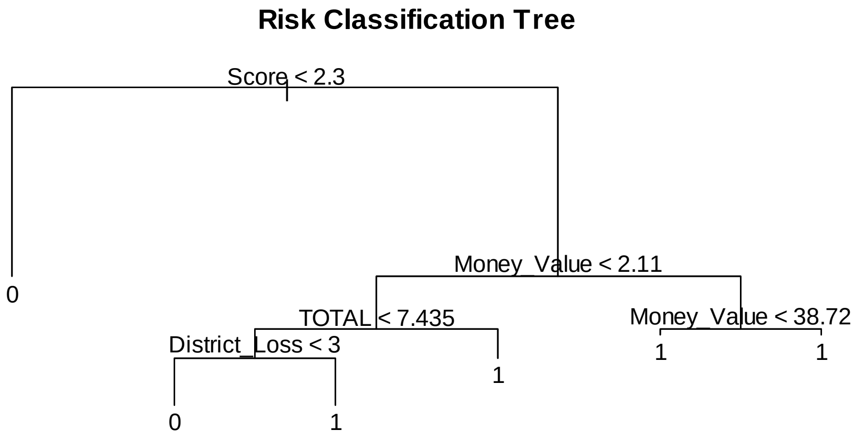


Problem 2: Audit Risk

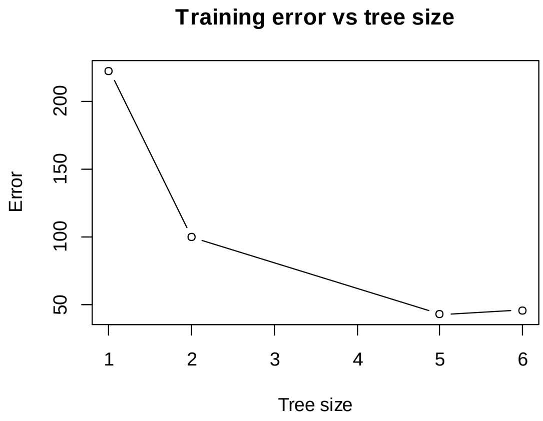
(1) Fit a classification tree to predict Risk using all variables with training dataset, the mis-classification error rate = 40/576 = **6.944%.** Test the performance with testing dataset, the confusion matrix is obtained as below.



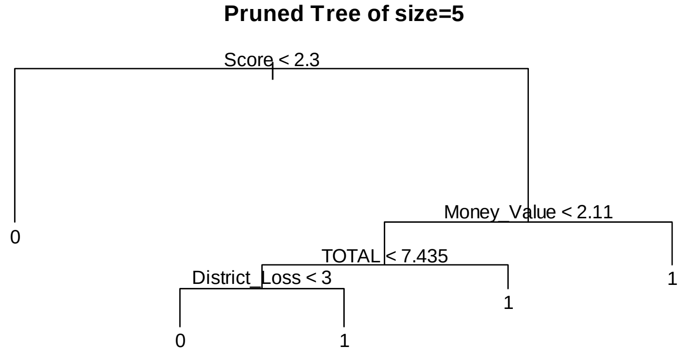
```
> library(tree)
> audit_train=read.csv("audit_train.csv", header=TRUE,sep=",")
> audit_test=read.csv("audit_test.csv", header=TRUE,sep=",")
> audit_train$Risk=as.factor(audit_train$Risk)
> fit.tree=tree(Risk~.,audit_train)
> summary(fit.tree)

Classification tree:
tree(formula = Risk ~ ., data = audit_train)
Variables actually used in tree construction:
[1] "Score"         "Money_Value"   "TOTAL"         "District_Loss"
Number of terminal nodes:  6 
Residual mean deviance:  0.486 = 277 / 570 
Misclassification error rate: 0.06944 = 40 / 576 
> par(mfrow=c(1,1))
> plot(fit.tree)
> text(fit.tree,pretty=0)
> title("Risk Classification Tree")
> preds=predict(fit.tree,audit_test,type="class")
> table(preds,audit_test$Risk)

preds   0   1
    0 106   5
    1   6  83
```



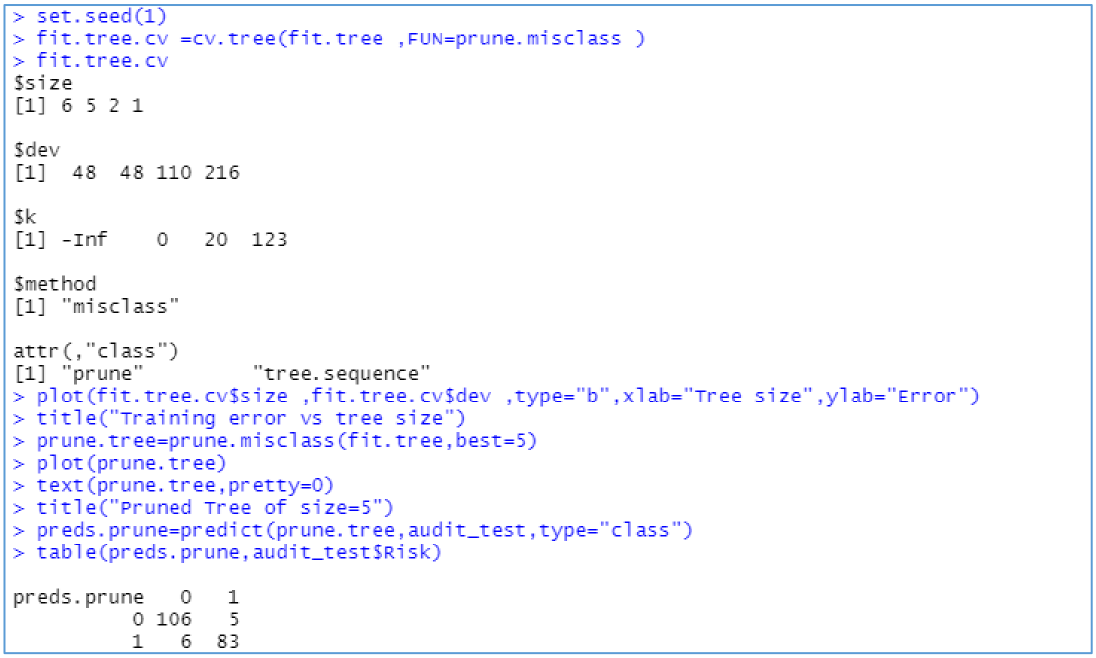
(2) Using CV to prune the tree, the plot of training error vs tree size is:



The tree with size 5 yields the best training error. The plot of pruned tree using size 5:



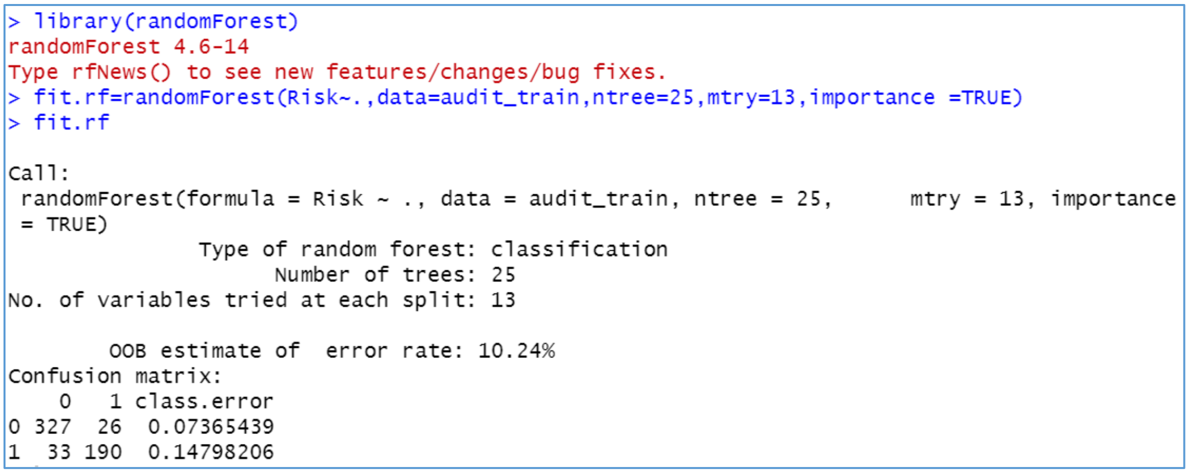
Code and confusion matrix:



```
> set.seed(1)
> fit.tree.cv =cv.tree(fit.tree ,FUN=prune.misclass )
> fit.tree.cv
$size
[1] 6 5 2 1

$dev
[1]  48  48 110 216

$k
[1] -Inf    0   20  123

$method
[1] "misclass"

attr(,"class")
[1] "prune"         "tree.sequence"
> plot(fit.tree.cv$size ,fit.tree.cv$dev ,type="b",xlab="Tree size",ylab="Error")
> title("Training error vs tree size")
> prune.tree=prune.misclass(fit.tree,best=5)
> plot(prune.tree)
> text(prune.tree,pretty=0)
> title("Pruned Tree of size=5")
> preds.prune=predict(prune.tree,audit_test,type="class")
> table(preds.prune,audit_test$Risk)

preds.prune   0   1
          0 106   5
          1   6  83
```

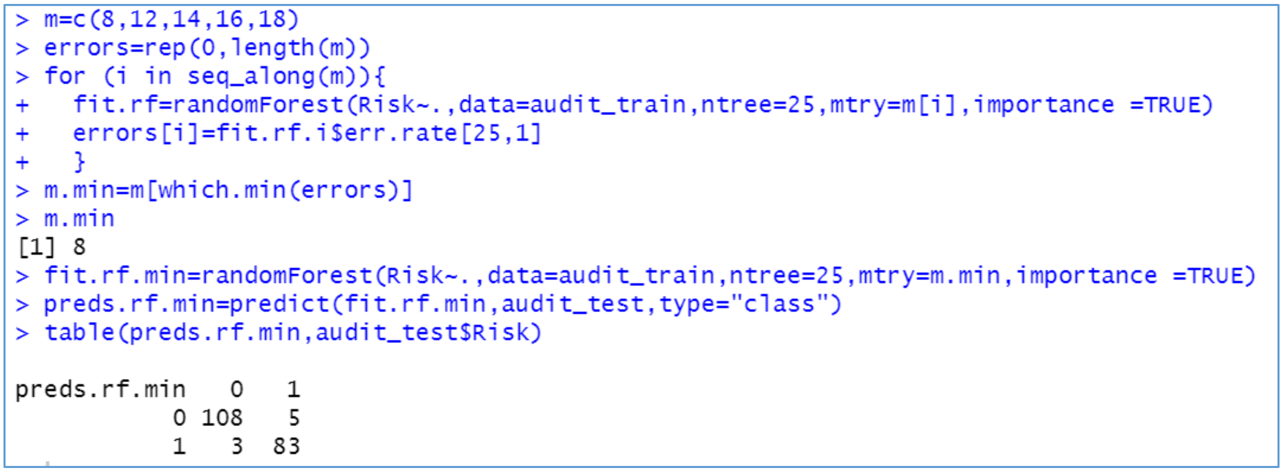
The test error = 11/200 = **5.5%**

(3) Using random forest with m=13 and ntree=25, the training error = (26+33)/576 = **10.24%**



```
> library(randomForest)
randomForest 4.6-14
Type rfNews() to see new features/changes/bug fixes.
> fit.rf=randomForest(Risk~.,data=audit_train,ntree=25,mtry=13,importance =TRUE)
> fit.rf

Call:
 randomForest(formula = Risk ~ ., data = audit_train, ntree = 25,      mtry = 13, importance
 = TRUE)
               Type of random forest: classification
                     Number of trees: 25
No. of variables tried at each split: 13

        OOB estimate of  error rate: 10.24%
Confusion matrix:
    0   1 class.error
0 327  26  0.07365439
1  33 190  0.14798206
```

(4) Repeating step 3 with different choices of m=8,12,14,16,18, the one with smallest mis-classification error on training dataset is m=8. The mis-classification error on testing dataset = 8/199 = <u>**4%**</u>



```
> m=c(8,12,14,16,18)
> errors=rep(0,length(m))
> for (i in seq_along(m)){
+   fit.rf=randomForest(Risk~.,data=audit_train,ntree=25,mtry=m[i],importance =TRUE)
+   errors[i]=fit.rf.i$err.rate[25,1]
+   }
> m.min=m[which.min(errors)]
> m.min
[1] 8
> fit.rf.min=randomForest(Risk~.,data=audit_train,ntree=25,mtry=m.min,importance =TRUE)
> preds.rf.min=predict(fit.rf.min,audit_test,type="class")
> table(preds.rf.min,audit_test$Risk)

preds.rf.min   0   1
           0 108   5
           1   3  83
```

(5) The training mis-classification error rate of Decision Tree was less than that using Radom Forest but Radom Forest performed better for the testing data. In general, the test error would be larger than the training error. However, in this problem, the test error is smaller than the training error regardless which method used. One more point to note is that one of the test record (record #61 with missing “Money_Value”) could not be classified using the Random Forest method.