```r
mydata <- read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")
set.seed(17)
## view the first few rows of the data
head(mydata)
```

```
##   admit gre  gpa rank
## 1     0 380 3.61    3
## 2     1 660 3.67    3
## 3     1 800 4.00    1
## 4     1 640 3.19    4
## 5     0 520 2.93    4
## 6     1 760 3.00    2
```

```r
summary(mydata)
```

```
##      admit             gre             gpa             rank
##  Min.   :0.0000   Min.   :220.0   Min.   :2.260   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:520.0   1st Qu.:3.130   1st Qu.:2.000
##  Median :0.0000   Median :580.0   Median :3.395   Median :2.000
##  Mean   :0.3175   Mean   :587.7   Mean   :3.390   Mean   :2.485
##  3rd Qu.:1.0000   3rd Qu.:660.0   3rd Qu.:3.670   3rd Qu.:3.000
##  Max.   :1.0000   Max.   :800.0   Max.   :4.000   Max.   :4.000
```

```r
## split training set and testing test
train_sub = sample(nrow(mydata),0.75*nrow(mydata))
train_data = mydata[train_sub,]
test_data = mydata[-train_sub,]
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.0.3
```

```
## randomForest 4.6-14
```

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

```r
## calculate mean of model's error rate based on OOB data
err<-as.numeric()
for(i in 1:((length(names(train_data)))-1)){
  print(i)
  mtry_test <- randomForest(as.factor(train_data$admit)~.,data=train_data,mtry=i,ntree=1000)
  err<- append( err, mean( mtry_test$err.rate ) )
}
```

```
## [1] 1
## [1] 2
## [1] 3
```

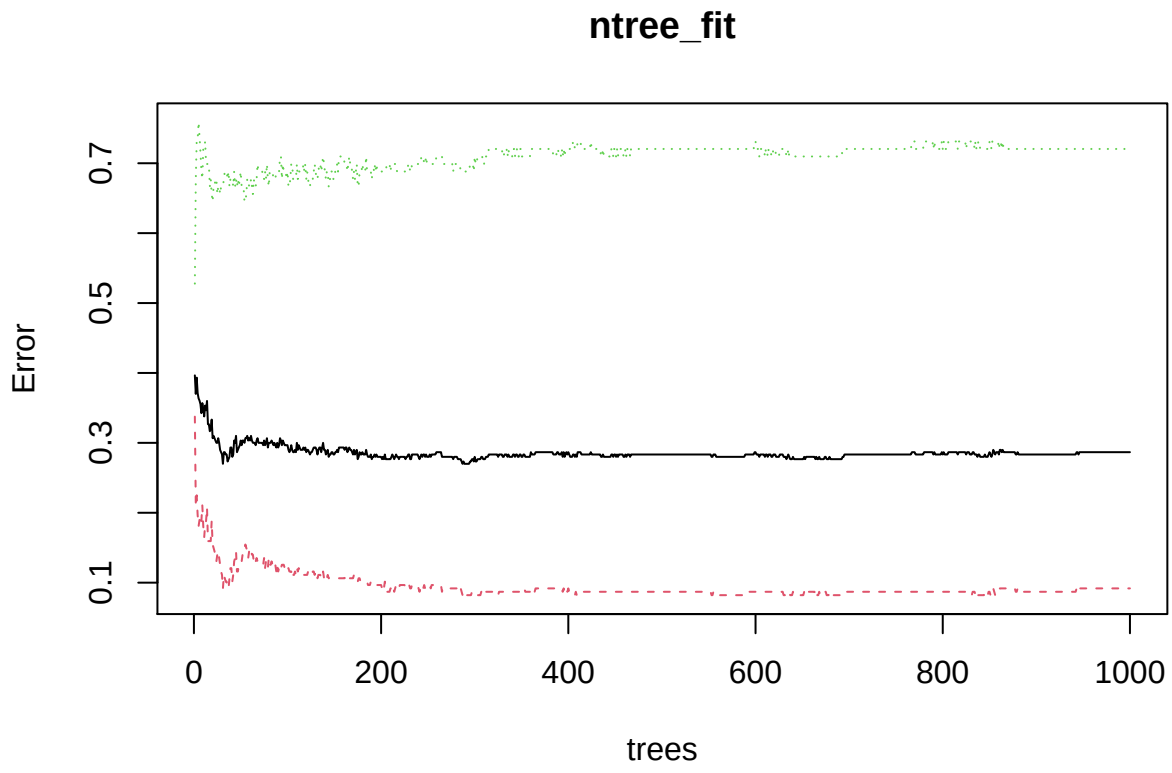
```r
print(err)
```

```
## [1] 0.3611373 0.3902456 0.4121190
```

```r
##obtain number of variables for a decision tree that result in the minimum error rate
mtry <- which.min(err)
print(mtry)
```

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

```r
ntree_fit <- randomForest(as.factor(train_data$admit)~., data=train_data, mtry=mtry, ntree=1000)
plot(ntree_fit)
```




```r
## then we get that when number of trees=900, the error is relatively stable
```

```r
rf<-randomForest(as.factor(train_data$admit)~., data=train_data, mtry=mtry, ntree=900, importance=T )
## show brief info about the random forest
print(rf)
```

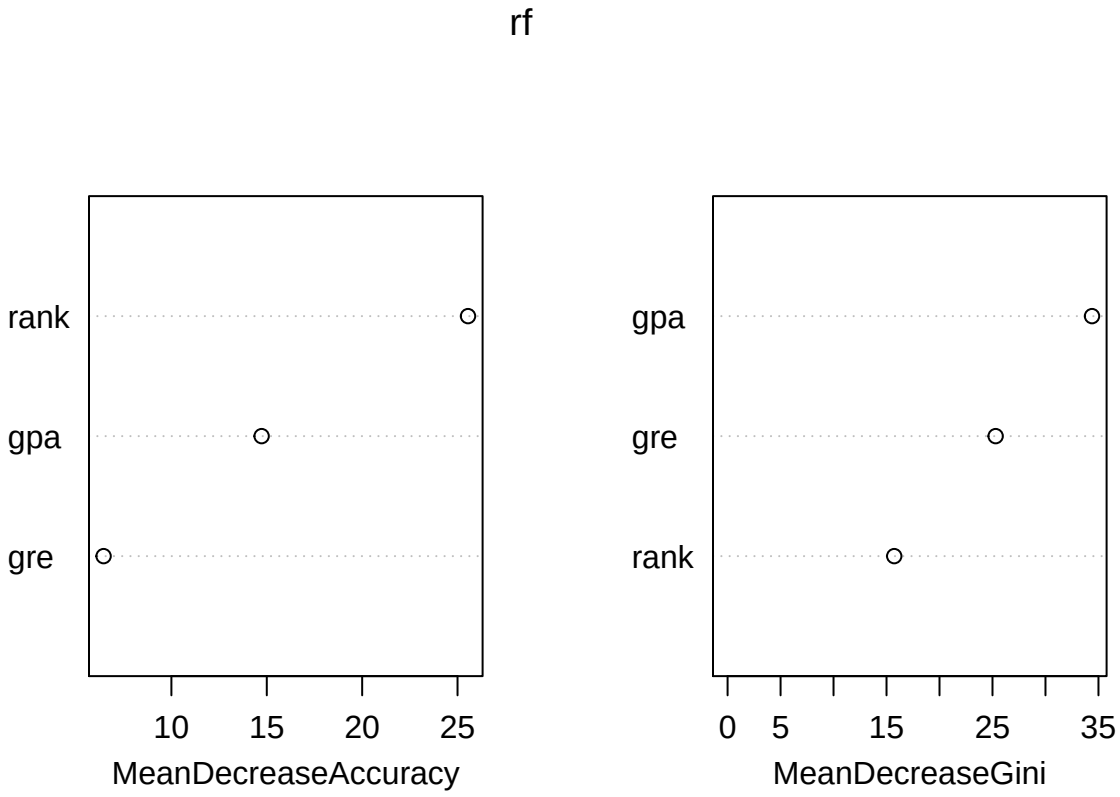
```
##
## Call:
##  randomForest(formula = as.factor(train_data$admit) ~ ., data = train_data,      mtry = mtry, ntree =
##                Type of random forest: classification
##                      Number of trees: 900
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 27.67%
```

```
## Confusion matrix:
##     0  1 class.error
## 0 190 17   0.0821256
## 1  66 27   0.7096774
```

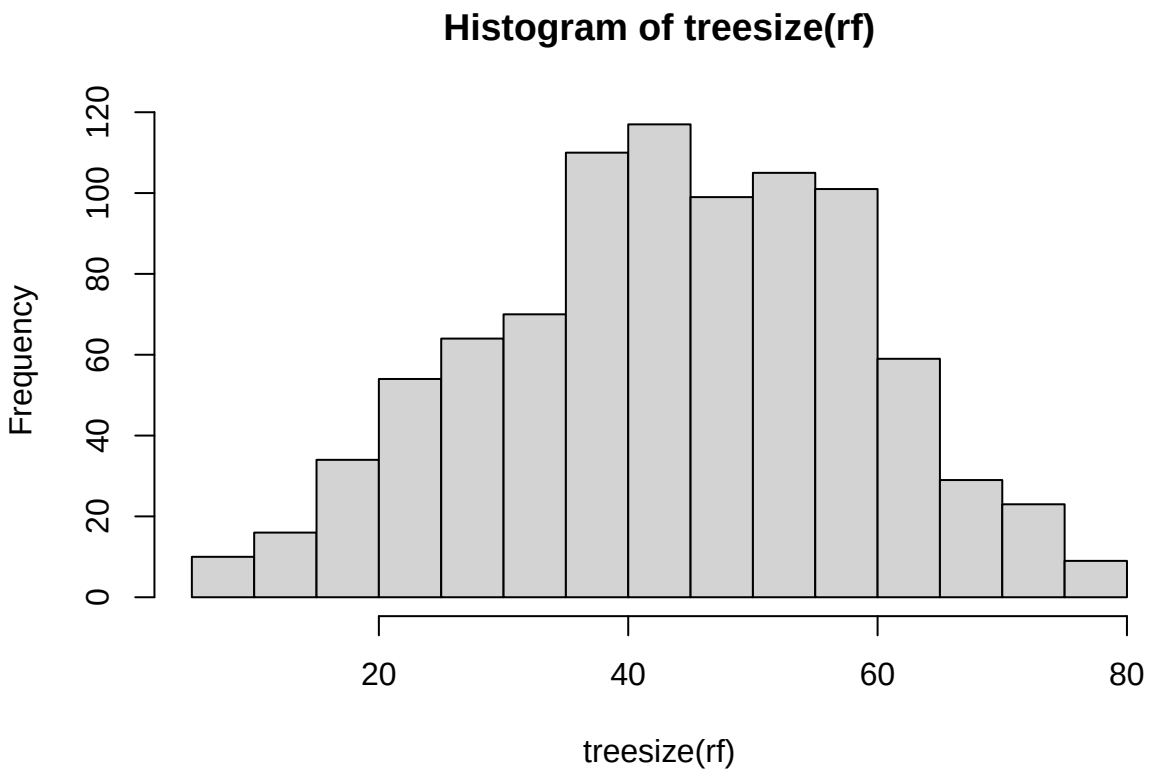
```
## show accuracy and gini coefficient
importance(rf)
```

```
##              0         1 MeanDecreaseAccuracy MeanDecreaseGini
## gre   5.781273  3.157630             6.443939         25.29761
## gpa  13.185219  8.552082            14.729268         34.39264
## rank 20.163610 19.448100            25.549313         15.72709
```

```
varImpPlot(rf)
```




```
## show number of nodes for each decision treee
hist(treesize(rf))
```




```
max(treesize(rf));min(treesize(rf))
```

```
## [1] 80
```

```
## [1] 5
```

```
pred1<-predict(rf,data=train_data)
Freq1<-table(pred1,train_data$admit)
## prediction accuracy
sum(diag(Freq1))/sum(Freq1)
```

```
## [1] 0.7233333
```