IncomePredictio.R

kavya

2019-04-21

```r
#install.packages("naniar")
library(naniar)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------------- tidyverse 1.2.1 -

## v ggplot2 3.1.0       v purrr   0.3.2
## v tibble  2.1.1       v dplyr   0.8.0.1
## v tidyr   0.8.3       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.4.0

## -- Conflicts ------------------------------------------------------------------ tidyverse_conflicts() -
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(GoodmanKruskal)
library(randomForest)
```

```
## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(ggplot2)
library(rpart)
library(rpart.plot)
library(party)
```

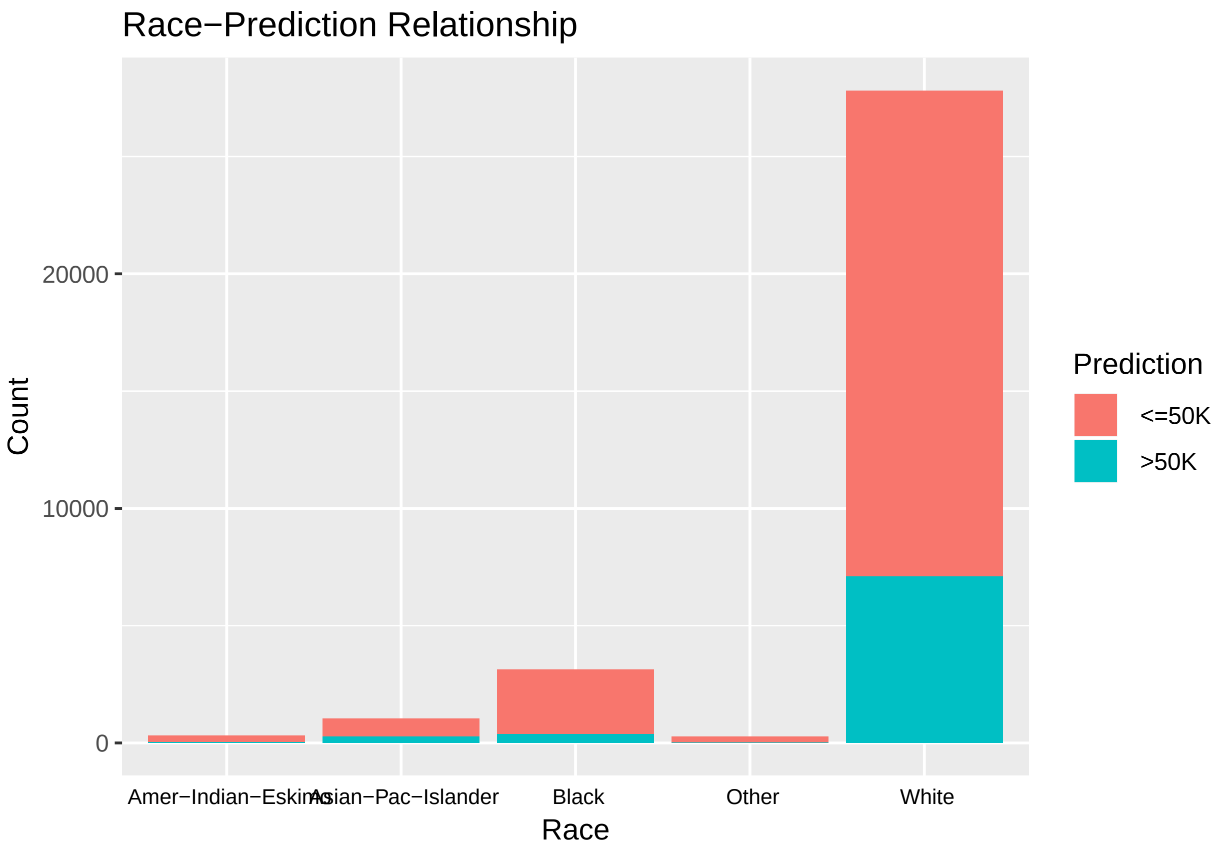
```
## Loading required package: grid

## Loading required package: mvtnorm

## Loading required package: modeltools

## Loading required package: stats4

## Loading required package: strucchange

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
```

```
##     as.Date, as.Date.numeric

## Loading required package: sandwich

##
## Attaching package: 'strucchange'

## The following object is masked from 'package:stringr':
##
##     boundary
```

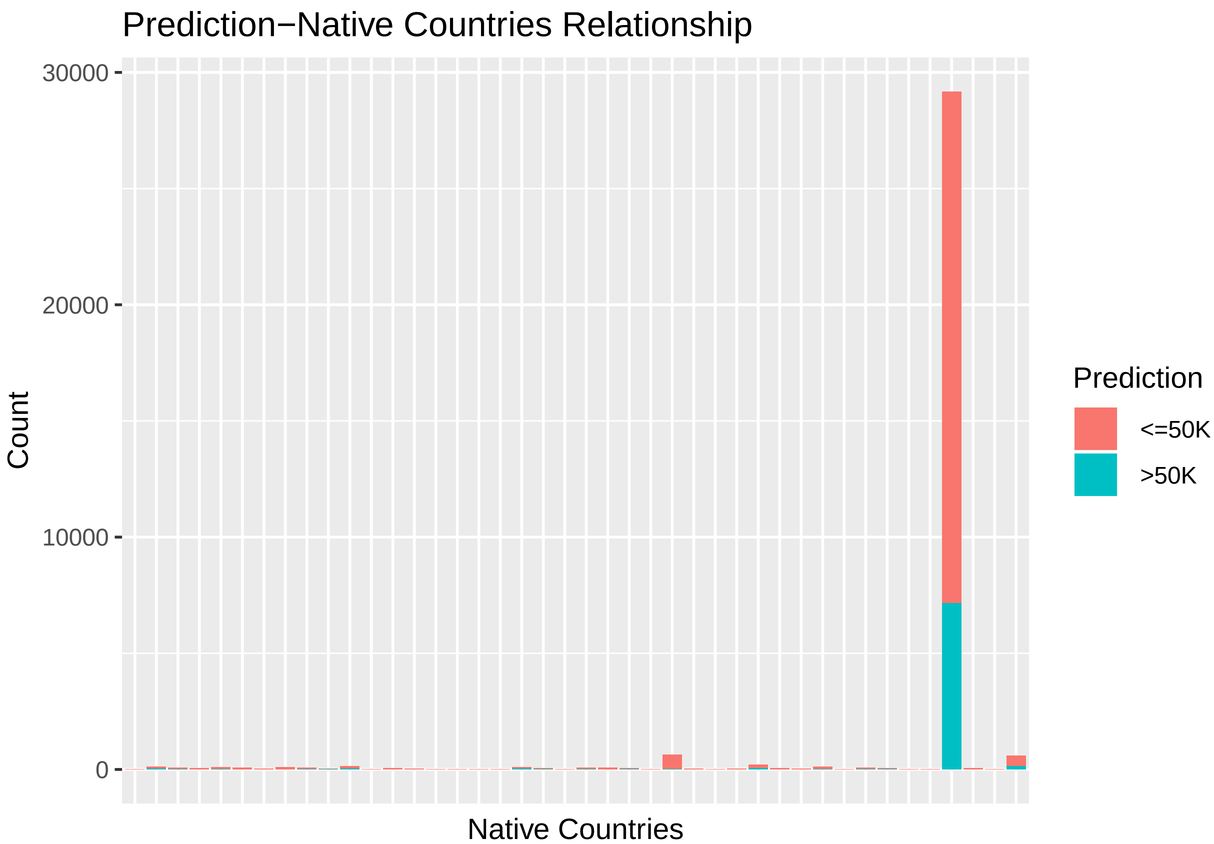
```r
library(e1071)

# Read the data into a data frame
dataset = read.table("adult.data",header = TRUE,sep = ",",na.strings = " ?")
dim(dataset)
```

```
## [1] 32561    15
```

```r
attach(dataset)
#Based on the initial analysis, the columns fnlwgt,race,capital.loss,native.country
#are dropped. Values stored in this column are skewed and do not contribute to any useful information

#fnlwgt is an attribute used in data generation during taking the census, it tells the instance belongs
#and provides no use for the defined tasks

#Skewed graph for Race attribute
ggplot(data.frame(dataset)) +
  geom_bar(aes(x=race,fill = as.factor(prediction)))+ ggtitle(label = "Race-Prediction Relationship")+
  labs(fill = "Prediction")  + xlab("Race")+ylab("Count")+
  theme(axis.text.x=element_text(color="black", size=8),
        axis.ticks.x=element_blank())
```



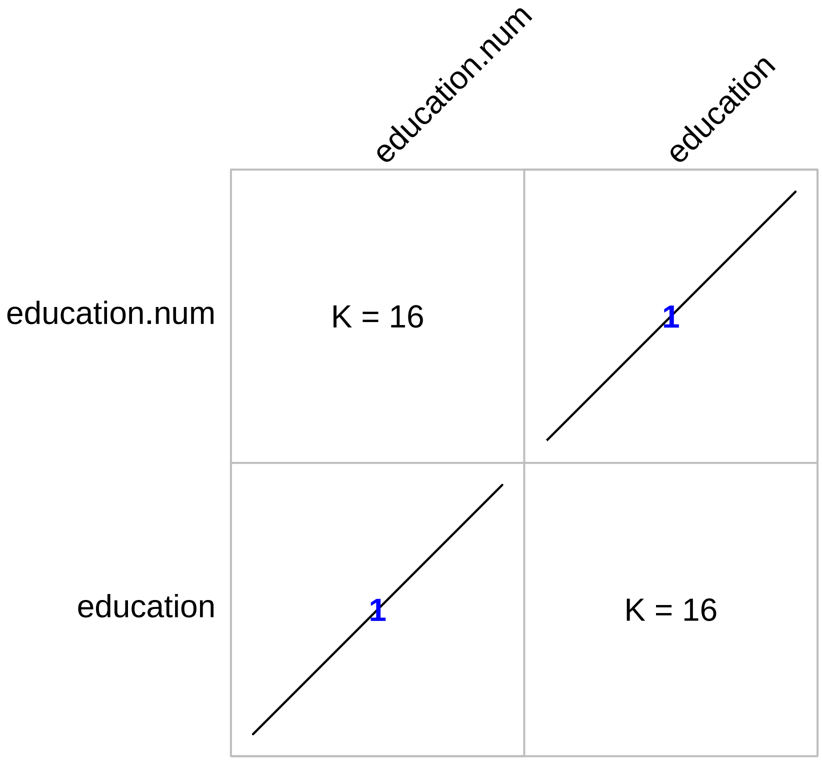

```
#Skewed graph for Native.countries attribute
ggplot(data.frame(dataset)) +
  geom_bar(aes(x=native.country,fill = as.factor(prediction)))+ ggtitle(label = "Prediction-Native Coun
  labs(fill = "Prediction")  + xlab("Native Countries")+ylab("Count")+
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```




```
#Hence we dropped the attribute fnlwgt, race and native.country
dataset = subset(dataset, select = -c(fnlwgt,race,native.country) )
dim(dataset)
```

```
## [1] 32561    12
```

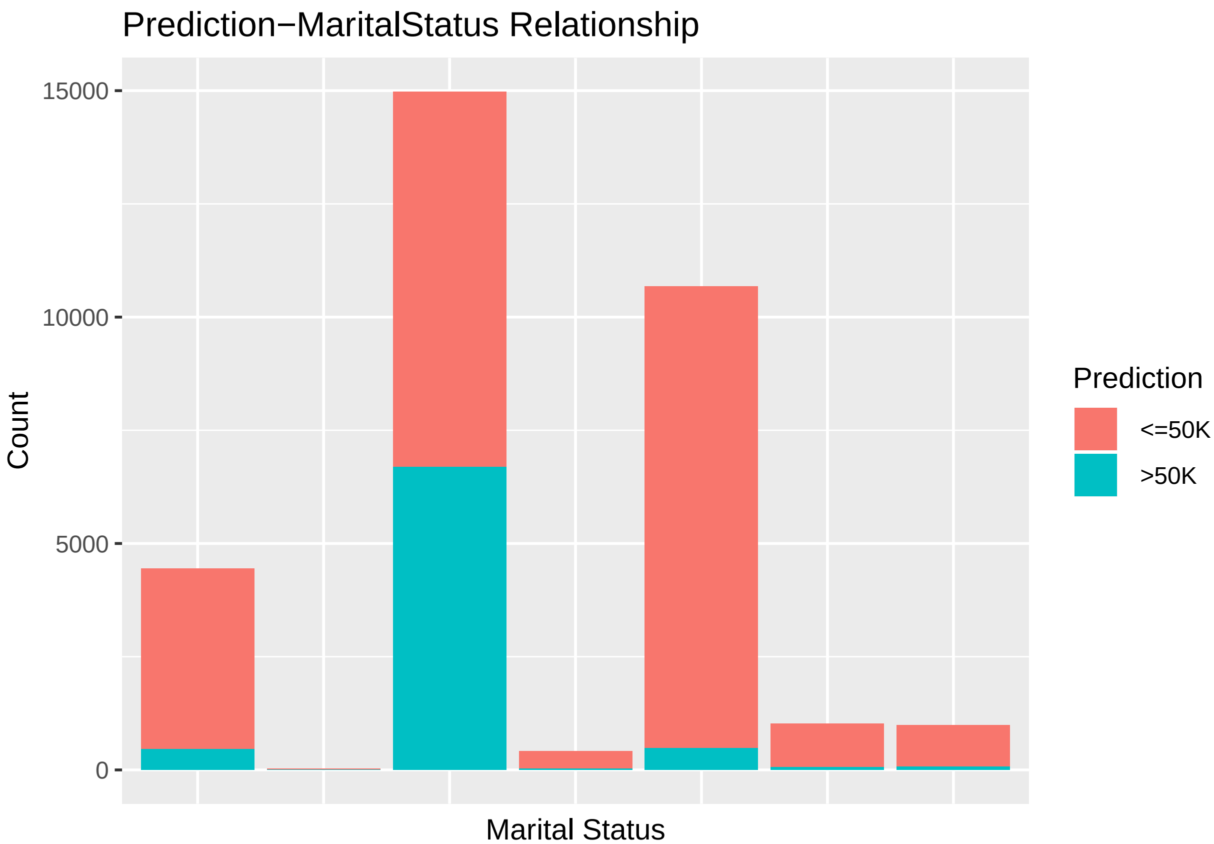
```
#As Education number and the "education" attribute are highly correlated, both signify the same
#thing. Hence "education.num" is dropped
varset1<- c("education.num","education")
datasetFrame1<- subset(dataset, select = varset1)
GKmatrix1<- GKtauDataframe(datasetFrame1)
plot(GKmatrix1, corrColors = "blue")
```




```r
#Redundant attributes "education.num" and capital loss are dropped
dataset = subset(dataset, select = -c(education.num,capital.loss) )
dim(dataset)
```

```
## [1] 32561    10
```

```r
#Skewed graph for Marital Status attribute
ggplot(data.frame(dataset)) +
  geom_bar(aes(x=marital.status,fill = as.factor(prediction)))+ ggtitle(label = "Prediction-MaritalStatu
  labs(fill = "Prediction")  + xlab("Marital Status")+ylab("Count")+
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```




```r
summary(dataset)
```

```
##        age                      workclass             education
##  Min.   :17.00    Private          :22696    HS-grad     :10501
##  1st Qu.:28.00    Self-emp-not-inc: 2541    Some-college: 7291
##  Median :37.00    Local-gov        : 2093    Bachelors   : 5355
##  Mean   :38.58    State-gov        : 1298    Masters     : 1723
##  3rd Qu.:48.00    Self-emp-inc     : 1116    Assoc-voc   : 1382
##  Max.   :90.00   (Other)           :  981    11th        : 1175
##                  NA's              : 1836   (Other)      : 5134
##                 marital.status             occupation
##   Divorced               : 4443    Prof-specialty : 4140
##   Married-AF-spouse      :   23    Craft-repair   : 4099
##   Married-civ-spouse     :14976    Exec-managerial: 4066
##   Married-spouse-absent:   418    Adm-clerical   : 3770
##   Never-married          :10683    Sales          : 3650
##   Separated              : 1025   (Other)         :10993
##   Widowed                :   993   NA's           : 1843
##           relationship          sex          capital.gain     hours.per.week
##   Husband        :13193    Female:10771   Min.   :    0    Min.   : 1.00
##   Not-in-family : 8305    Male  :21790   1st Qu.:    0    1st Qu.:40.00
##   Other-relative:   981                   Median :    0    Median :40.00
##   Own-child      : 5068                   Mean   : 1078    Mean   :40.44
##   Unmarried      : 3446                   3rd Qu.:    0    3rd Qu.:45.00
##   Wife           : 1568                   Max.   :99999    Max.   :99.00
##
##   prediction
```

```
##   <=50K:24720
##   >50K : 7841
##
##
##
##
##
```

```r
#Missing value analysis
# total number of rows with NA value
sum(is.na(dataset))
```

```
## [1] 3679
```

```r
# find the number of null values for each attribute
row = sapply(dataset,  function(x)
  sum(is.na(x)))

row = data.frame(row)
print(row)
```

```
##                  row
## age                0
## workclass       1836
## education          0
## marital.status     0
## occupation      1843
## relationship       0
## sex                0
## capital.gain       0
## hours.per.week     0
## prediction         0
```

```r
# find only those instances where workclass is null
#d1 <- filter(dataset, is.na("workclass"))
#summary(d1)
#head(d1)

# found out that whenever the value of workclass is missing then the value of occupation is also missin
# this suggest some co-relation between them.

#replace the NA of WORKCLASS WITH "Unknown".
dataset$workclass <- as.character(dataset$workclass)
dataset$workclass[is.na(dataset$workclass)] <- "Unknown"
dataset$workclass <- factor(dataset$workclass)

dataset$occupation <- as.character(dataset$occupation)
dataset$occupation[is.na(dataset$occupation)] <- "Unknown"
dataset$occupation <- factor(dataset$occupation)
dim(dataset)
```

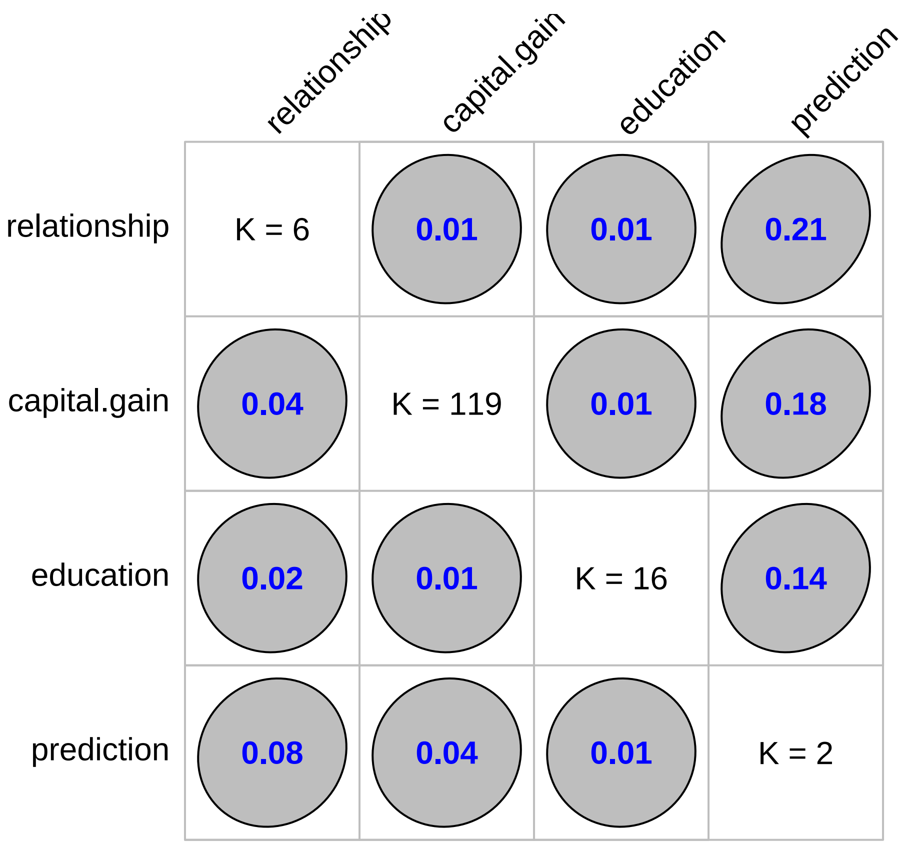
```
## [1] 32561    10
```

```r
names(dataset)
```

```
##  [1] "age"            "workclass"      "education"      "marital.status"
##  [5] "occupation"     "relationship"   "sex"            "capital.gain"
```

```
##  [9] "hours.per.week" "prediction"
```

```r
#Our data after preprocessing consists of just 10 columns
# attribute importance based on correlation
varset1<- c("relationship","capital.gain","education", "prediction")
datasetFrame1<- subset(dataset, select = varset1)
GKmatrix1<- GKtauDataframe(datasetFrame1)
plot(GKmatrix1, corrColors = "blue")
```




```r
#Preparing Test data by applying the preprocessing steps applied to the training data above
test_data = read.table("adult.test",header = TRUE,sep = ",",na.strings = " ?")
dim(test_data)
```

```
## [1] 16281    15
```

```r
attach(test_data)
```

```
## The following objects are masked from dataset:
##
##     age, capital.gain, capital.loss, education, education.num,
##     fnlwgt, hours.per.week, marital.status, native.country,
##     occupation, prediction, race, relationship, sex, workclass
```

```r
#Dropping redundant columns
test_data = subset(test_data, select = -c(fnlwgt,race,native.country,education.num,capital.loss) )
dim(test_data)
```

```
## [1] 16281    10
```

```r
#Missing value analysis
# total number of rows with NA value
```

```r
sum(is.na(test_data))
```

```
## [1] 1929
```

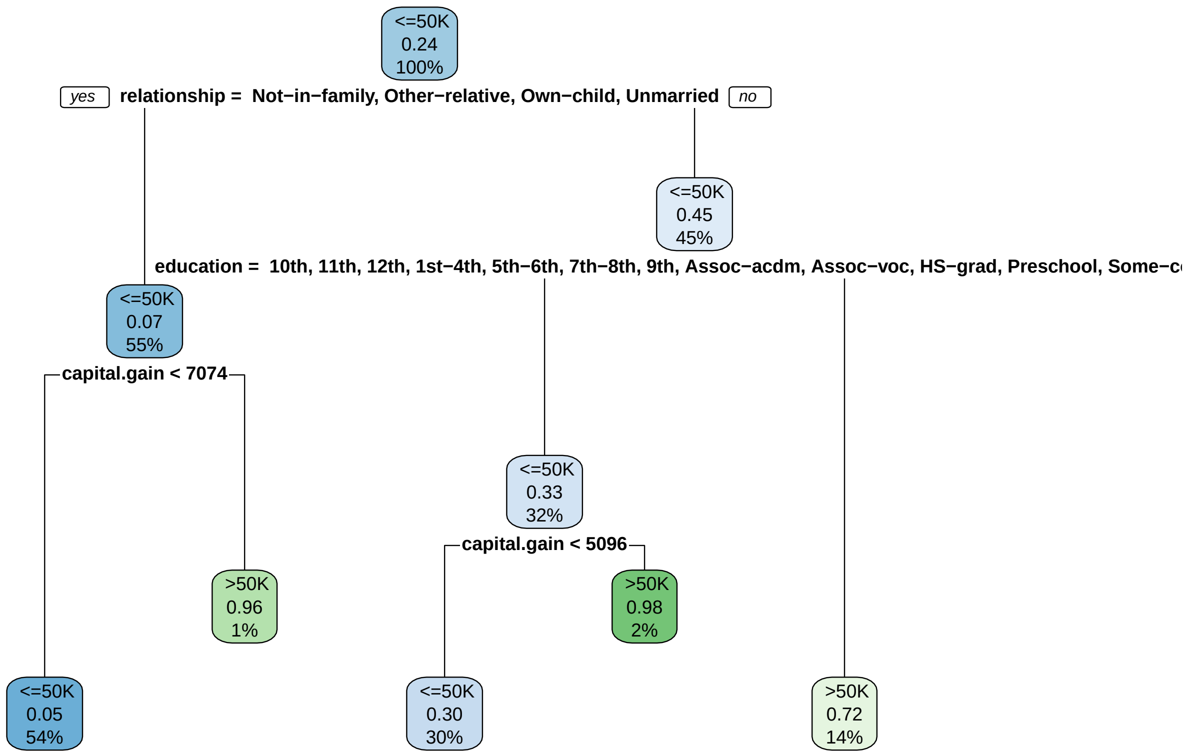
```r
# find the number of null values for each attribute
row = sapply(test_data,  function(x)
  sum(is.na(x)))

row = data.frame(row)

# find only those instances where workclass is null
#d1 <- filter(dataset, is.na("workclass"))
#summary(d1)
#head(d1)

# found out that whenever the value of workclass is missing then the value of occupation is also missin
# this suggest some co-relation between them.

#replace the NA of WORKCLASS WITH "Unknown".
test_data$workclass <- as.character(test_data$workclass)
test_data$workclass[is.na(test_data$workclass)] <- "Unknown"
test_data$workclass <- factor(test_data$workclass)

test_data$occupation <- as.character(test_data$occupation)
test_data$occupation[is.na(test_data$occupation)] <- "Unknown"
test_data$occupation <- factor(test_data$occupation)
dim(test_data)
```

```
## [1] 16281    10
```

```r
names(test_data)
```

```
##  [1] "age"            "workclass"      "education"      "marital.status"
##  [5] "occupation"     "relationship"   "sex"            "capital.gain"
##  [9] "hours.per.week" "prediction"
```

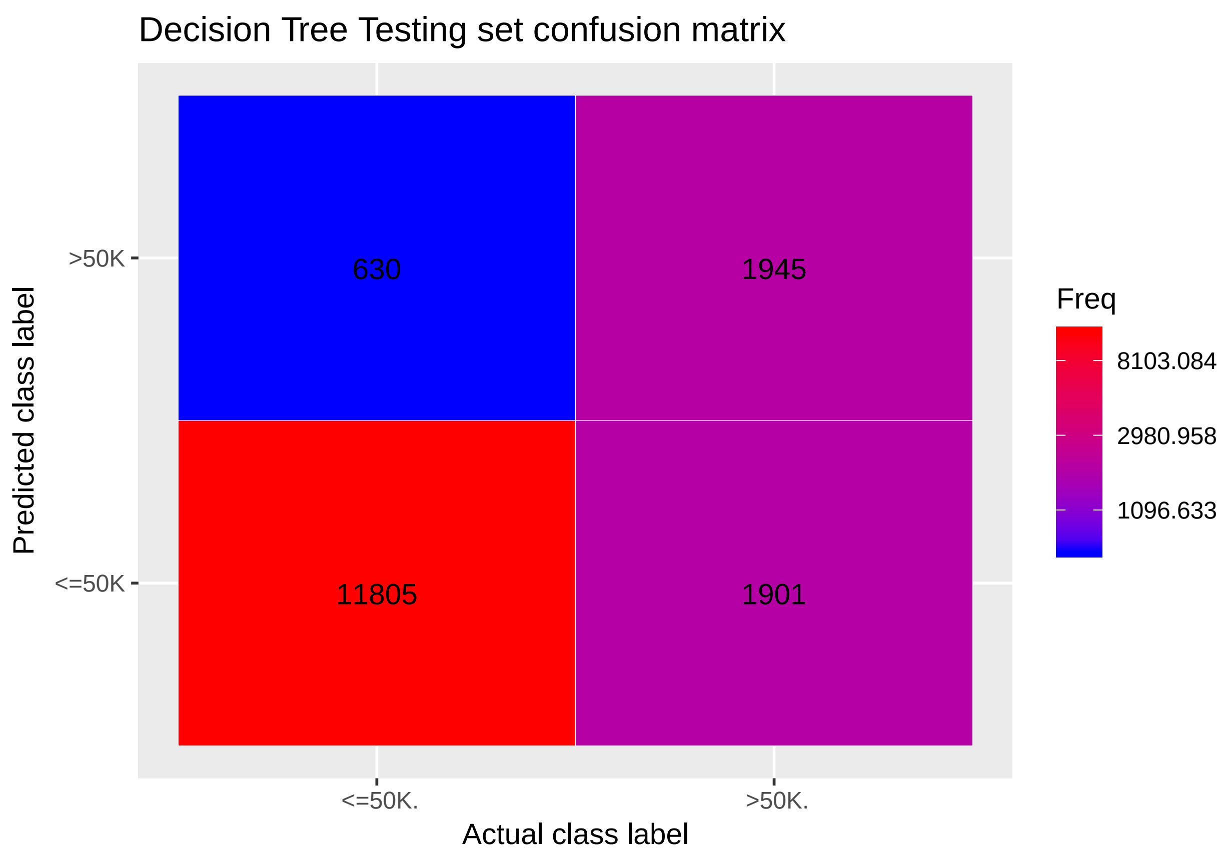
```r
# Decision tree creation based on training dataset
dtree <- rpart(prediction ~ ., data = dataset, method = 'class', model = TRUE)
rpart.plot(dtree)
```




```r
val_predicted <- predict(dtree, dataset, type = "class")

confMatrix <- (table(dataset$prediction, val_predicted))
print(confMatrix)
```

```
##         val_predicted
##           <=50K  >50K
##    <=50K  23473  1247
##    >50K    3816  4025
```

```r
# given error
#accuracy <- sum(diag(confMatrix))/sum(confMatrix)
#print(accuracy)

# run the model on test data
val_predicted <- predict(dtree, test_data, type = "class")
confMatrix <- as.data.frame(table(test_data$prediction, val_predicted))

ggplot(data =  confMatrix, mapping = aes(x = Var1, y = val_predicted)) +
  ggtitle("Decision Tree Testing set confusion matrix")+
  geom_tile(aes(fill = Freq), colour = "white") +
  xlab("Actual class label")+
  ylab("Predicted class label")+
  geom_text(aes(label = sprintf("%1.0f", Freq)), vjust = 1) +
  scale_fill_gradient(low = "blue",
                      high = "red",
                      trans = "log")
```




```r
print(confMatrix)
```

```
##      Var1 val_predicted  Freq
## 1  <=50K.          <=50K 11805
## 2   >50K.          <=50K  1901
## 3  <=50K.           >50K   630
## 4   >50K.           >50K  1945
```

```r
confMatrix <- (table(test_data$prediction, val_predicted))
accuracy <- sum(diag(confMatrix))/sum(confMatrix)
print(accuracy)
```

```
## [1] 0.8445427
```

```r
# Build Naive Bayes Model
model <- naiveBayes(prediction ~ ., data = dataset)
print(model)
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##     <=50K      >50K
## 0.7591904 0.2408096
##
```

```
## Conditional probabilities:
##        age
## Y           [,1]     [,2]
##   <=50K 36.78374 14.02009
##   >50K  44.24984 10.51903
##
##        workclass
## Y         Federal-gov    Local-gov  Never-worked      Private
##   <=50K 0.0238268608 0.0597087379  0.0002831715 0.7173543689
##   >50K  0.0473153934 0.0786889427  0.0000000000 0.6329549802
##        workclass
## Y         Self-emp-inc  Self-emp-not-inc    State-gov  Without-pay
##   <=50K  0.0199838188      0.0735032362 0.0382281553 0.0005663430
##   >50K   0.0793266165      0.0923351613 0.0450197679 0.0000000000
##        workclass
## Y             Unknown
##   <=50K 0.0665453074
##   >50K  0.0243591379
##
##        education
## Y               10th         11th         12th      1st-4th      5th-6th
##   <=50K 0.0352346278 0.0451051780 0.0161812298 0.0065533981 0.0128236246
##   >50K  0.0079071547 0.0076520852 0.0042086469 0.0007652085 0.0020405561
##        education
## Y            7th-8th          9th   Assoc-acdm    Assoc-voc    Bachelors
##   <=50K 0.0245145631 0.0197006472 0.0324433657 0.0413025890 0.1267799353
##   >50K  0.0051013901 0.0034434383 0.0337967096 0.0460400459 0.2832546869
##        education
## Y          Doctorate      HS-grad      Masters    Preschool  Prof-school
##   <=50K 0.0043284790 0.3570388350 0.0309061489 0.0020631068 0.0061893204
##   >50K  0.0390256345 0.2136207116 0.1223058283 0.0000000000 0.0539472006
##        education
## Y         Some-college
##   <=50K  0.2388349515
##   >50K   0.1768907027
##
##        marital.status
## Y          Divorced  Married-AF-spouse  Married-civ-spouse
##   <=50K 0.161003236        0.000525890         0.335113269
##   >50K  0.059048591        0.001275348         0.853462569
##        marital.status
## Y         Married-spouse-absent  Never-married   Separated     Widowed
##   <=50K            0.015533981    0.412297735 0.038794498 0.036731392
##   >50K             0.004336182    0.062619564 0.008417294 0.010840454
##
##        occupation
## Y         Adm-clerical  Armed-Forces  Craft-repair  Exec-managerial
##   <=50K  0.1319983819  0.0003236246  0.1282362460     0.0848705502
##   >50K   0.0646601199  0.0001275348  0.1184797857     0.2509883943
##        occupation
## Y         Farming-fishing  Handlers-cleaners  Machine-op-inspct
##   <=50K     0.0355582524       0.0519417476       0.0708737864
##   >50K      0.0146664966       0.0109679888       0.0318836883
##        occupation
```

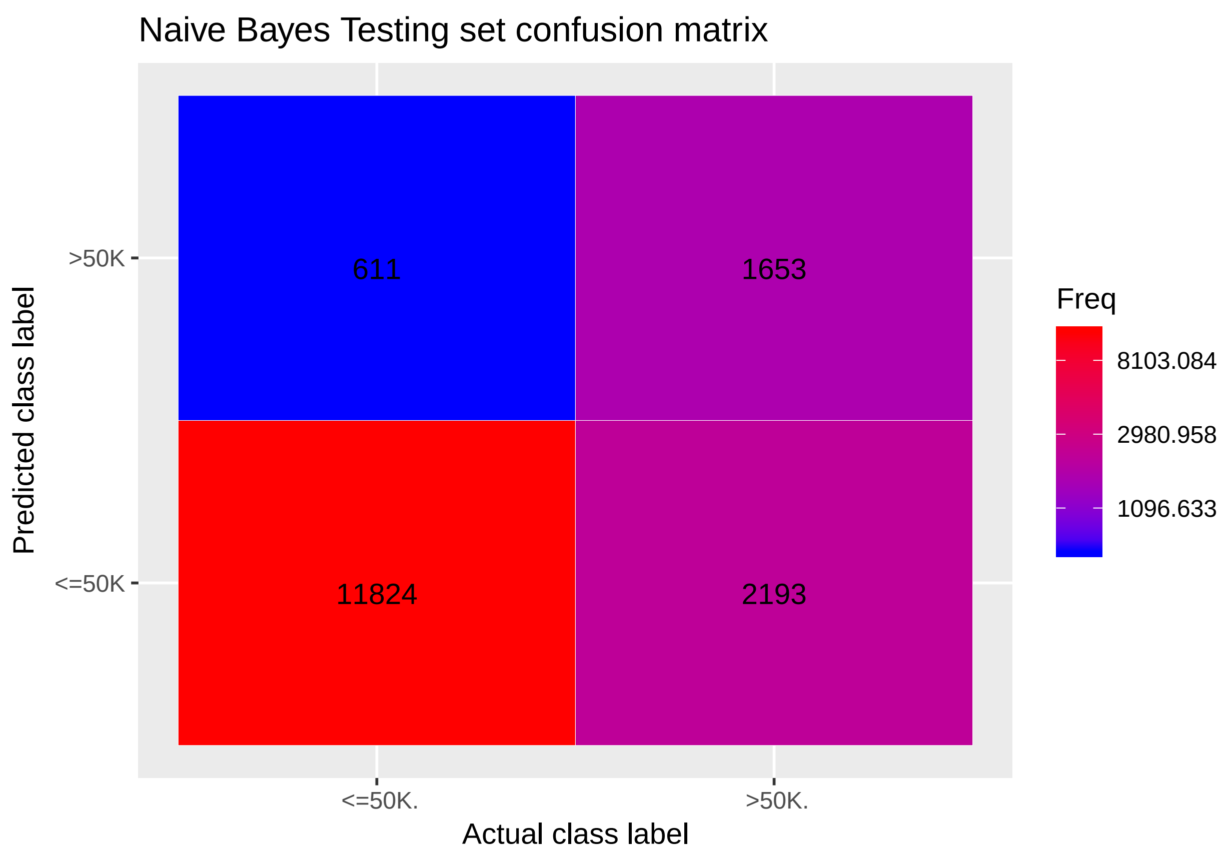
```
## Y          Other-service  Priv-house-serv  Prof-specialty  Protective-serv
##    <=50K   0.1277508091     0.0059870550    0.0922734628     0.0177184466
##    >50K    0.0174722612     0.0001275348    0.2370871062     0.0269098329
##        occupation
## Y                 Sales  Tech-support  Transport-moving      Unknown
##    <=50K 0.1078883495  0.0260922330      0.0516585761 0.0668284790
##    >50K  0.1253666624  0.0360923352      0.0408111210 0.0243591379
##
##        relationship
## Y             Husband  Not-in-family  Other-relative   Own-child
##    <=50K 0.294296117    0.301334951     0.038187702 0.202305825
##    >50K  0.754750670    0.109169749     0.004718786 0.008544828
##        relationship
## Y           Unmarried        Wife
##    <=50K 0.130582524 0.033292880
##    >50K  0.027802576 0.095013391
##
##        sex
## Y            Female      Male
##    <=50K 0.3880259 0.6119741
##    >50K  0.1503635 0.8496365
##
##        capital.gain
## Y              [,1]       [,2]
##    <=50K  148.7525   963.1393
##    >50K  4006.1425 14570.3790
##
##        hours.per.week
## Y             [,1]     [,2]
##    <=50K 38.84021 12.31899
##    >50K  45.47303 11.01297
```

```r
# Test model on training data
vals_predicted <- predict(model, newdata = dataset)
confMatrix <- table(dataset$prediction, vals_predicted)

# Prints confusion matrix indicating number of values correctly predicted and not

print(confMatrix)
```

```
##        vals_predicted
##          <=50K  >50K
##    <=50K  23511  1209
##    >50K    4387  3454
```

```r
#accuracy <- sum(diag(confMatrix))/sum(confMatrix)
#print(accuracy)




# Test model on test data
vals_predicted <- predict(model, newdata = test_data)
confMatrix <- as.data.frame(table(test_data$prediction, vals_predicted))

ggplot(data =  confMatrix, mapping = aes(x = Var1, y = vals_predicted)) +
```

```
  ggtitle("Naive Bayes Testing set confusion matrix")+
  geom_tile(aes(fill = Freq), colour = "white") +
  xlab("Actual class label")+
  ylab("Predicted class label")+
  geom_text(aes(label = sprintf("%1.0f", Freq)), vjust = 1) +
  scale_fill_gradient(low = "blue",
                      high = "red",
                      trans = "log")
```

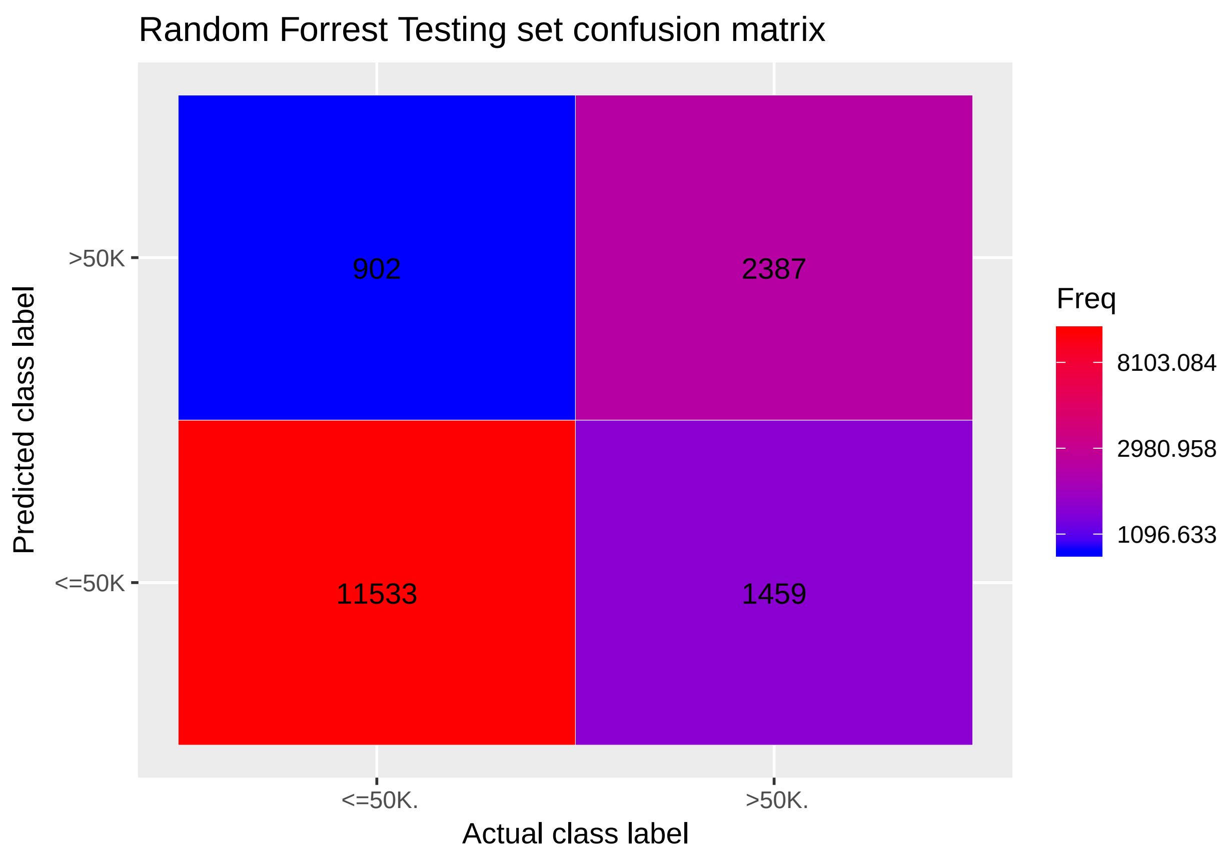



```
# Prints confusion matrix indicating number of values correctly predicted and not
confMatrix <- (table(test_data$prediction, vals_predicted))
accuracy <- sum(diag(confMatrix))/sum(confMatrix)
print(accuracy)
```

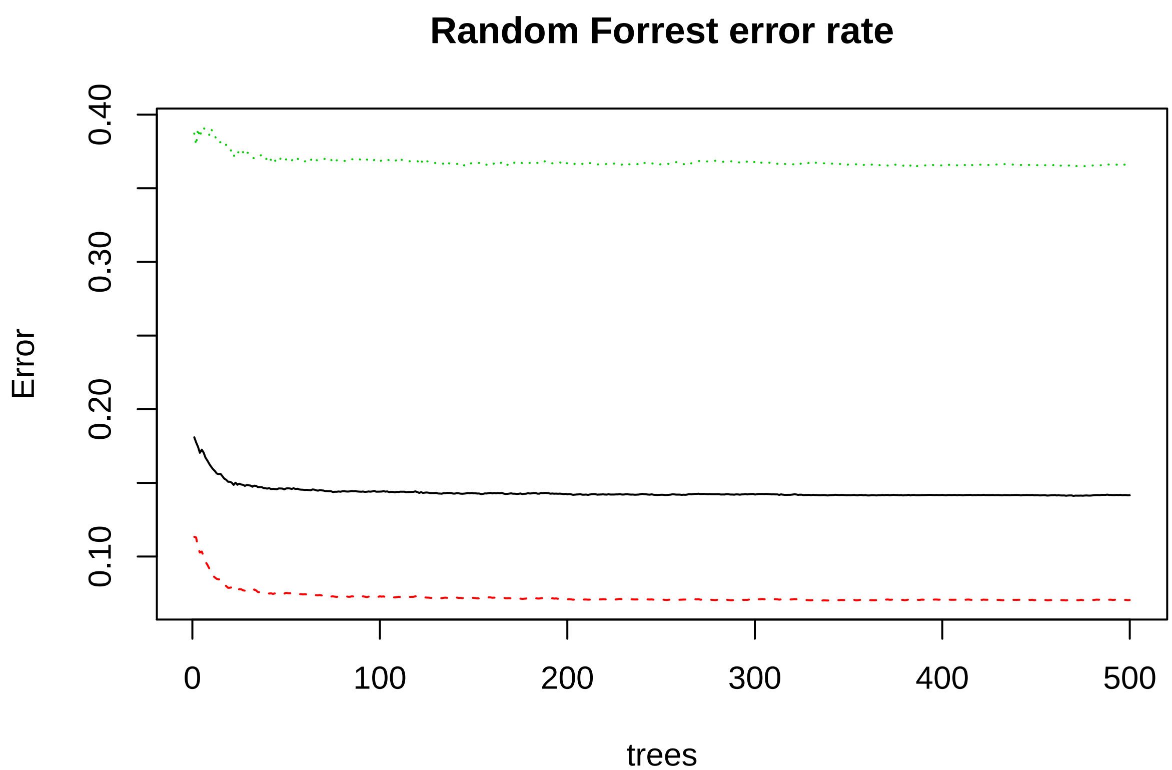
```
## [1] 0.8277747
```

```
# Build a Random Forrest
dtree <- randomForest(prediction ~ ., data = dataset)
val_predicted <- predict(dtree, dataset, type = 'response')
confMatrix <- (table(dataset$prediction, val_predicted))

# Plots error rate with respect to increase in number of trees generated
#plot(dtree,main="Random Forrest error rate")
#accuracy <- sum(diag(confMatrix))/sum(confMatrix)
#print(accuracy)

# On testing data
```

```r
val_predicted <- predict(dtree, test_data, type = 'response')
confMatrix <- as.data.frame(table(test_data$prediction, val_predicted))
ggplot(data =  confMatrix, mapping = aes(x = Var1, y = val_predicted)) +
  ggtitle("Random Forrest Testing set confusion matrix")+
  geom_tile(aes(fill = Freq), colour = "white") +
  xlab("Actual class label")+
  ylab("Predicted class label")+
  geom_text(aes(label = sprintf("%1.0f", Freq)), vjust = 1) +
  scale_fill_gradient(low = "blue",
                      high = "red",
                      trans = "log")
```




```r
# Plots error rate with respect to increase in number of trees generated
plot(dtree,main="Random Forrest error rate")
```




```r
confMatrix <- (table(test_data$prediction, val_predicted))
accuracy <- sum(diag(confMatrix))/sum(confMatrix)
print(accuracy)
```

```
## [1] 0.8549843
```