CKME - BANK MARKETING DATASET

Installation of Packages and adding it to library

install.packages("ggplot2") install.packages("corrplot") install.packages("caret") install.packages("dplyr") install.packages("caTools") install.packages("faraway") install.packages("modelr") install.packages("ROCR") install.packages("randomForest") install.packages("h2o") install.packages("e1071")

```
library(ggplot2)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(caTools)
library(faraway)
```

```
##
## Attaching package: 'faraway'
```

```
## The following object is masked from 'package:lattice':
##
##     melanoma
```

```
library(modelr)
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
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
```

```
library(h2o)
```

```
##
## ----------------------------------------------------------------------
##
## Your next step is to start H2O:
##     > h2o.init()
##
## For H2O package documentation, ask for help:
##     > ??h2o
##
## After starting H2O, you can use the Web UI at http://localhost:54321
## For more information visit http://docs.h2o.ai
##
## ----------------------------------------------------------------------
##
## Attaching package: 'h2o'
## The following objects are masked from 'package:stats':
##
##     cor, sd, var
## The following objects are masked from 'package:base':
##
##     %*%, %in%, &&, ||, apply, as.factor, as.numeric, colnames,
##     colnames<-, ifelse, is.character, is.factor, is.numeric, log,
##     log10, log1p, log2, round, signif, trunc
```

```
library(e1071)
```

set working directory

```
setwd("C:/Users/iss/Desktop/Ryerson/CKME136/BANK")
```

Read CSV file

Dataset is obtained from UCI.edu related to European banks marketing campaign carried for term deposits.

```
bank = read.csv("bank-additional-full.csv",sep=";", header=T)
```

Summary of the dataset

```
summary(bank)
```

```
##       age                 job            marital
##  Min.   :17.00   admin.     :10422   divorced: 4612
##  1st Qu.:32.00   blue-collar: 9254   married :24928
##  Median :38.00   technician : 6743   single  :11568
##  Mean   :40.02   services   : 3969   unknown :   80
##  3rd Qu.:47.00   management : 2924
##  Max.   :98.00   retired    : 1720
##                  (Other)    : 6156
##                education        default         housing
##  university.degree  :12168   no     :32588   no     :18622
##  high.school        : 9515   unknown: 8597   unknown:  990
##  basic.9y           : 6045   yes    :    3   yes    :21576
##  professional.course: 5243
##  basic.4y           : 4176
##  basic.6y           : 2292
##  (Other)            : 1749
##       loan            contact          month       day_of_week
##  no     :33950   cellular :26144   may    :13769   fri:7827
##  unknown:  990   telephone:15044   jul    : 7174   mon:8514
##  yes    : 6248                     aug    : 6178   thu:8623
##                                    jun    : 5318   tue:8090
##                                    nov    : 4101   wed:8134
##                                    apr    : 2632
##                                    (Other): 2016
##     duration         campaign          pdays          previous
##  Min.   :   0.0   Min.   : 1.000   Min.   :  0.0   Min.   :0.000
##  1st Qu.: 102.0   1st Qu.: 1.000   1st Qu.:999.0   1st Qu.:0.000
##  Median : 180.0   Median : 2.000   Median :999.0   Median :0.000
##  Mean   : 258.3   Mean   : 2.568   Mean   :962.5   Mean   :0.173
##  3rd Qu.: 319.0   3rd Qu.: 3.000   3rd Qu.:999.0   3rd Qu.:0.000
##  Max.   :4918.0   Max.   :56.000   Max.   :999.0   Max.   :7.000
##
##         poutcome      emp.var.rate      cons.price.idx  cons.conf.idx
##  failure    : 4252   Min.   :-3.40000   Min.   :92.20   Min.   :-50.8
##  nonexistent:35563   1st Qu.:-1.80000   1st Qu.:93.08   1st Qu.:-42.7
##  success    : 1373   Median : 1.10000   Median :93.75   Median :-41.8
##                      Mean   : 0.08189   Mean   :93.58   Mean   :-40.5
##                      3rd Qu.: 1.40000   3rd Qu.:93.99   3rd Qu.:-36.4
##                      Max.   : 1.40000   Max.   :94.77   Max.   :-26.9
##
##    euribor3m      nr.employed     y
##  Min.   :0.634   Min.   :4964   no :36548
##  1st Qu.:1.344   1st Qu.:5099   yes: 4640
##  Median :4.857   Median :5191
##  Mean   :3.621   Mean   :5167
##  3rd Qu.:4.961   3rd Qu.:5228
##  Max.   :5.045   Max.   :5228
##
```

```r
head(bank)
```

```
##   age       job marital   education default housing loan   contact month
## 1  56 housemaid married    basic.4y      no      no   no telephone   may
## 2  57  services married high.school unknown      no   no telephone   may
## 3  37  services married high.school      no     yes   no telephone   may
## 4  40    admin. married    basic.6y      no      no   no telephone   may
## 5  56  services married high.school      no      no  yes telephone   may
## 6  45  services married    basic.9y unknown      no   no telephone   may
##   day_of_week duration campaign pdays previous    poutcome emp.var.rate
## 1         mon      261        1   999        0 nonexistent          1.1
## 2         mon      149        1   999        0 nonexistent          1.1
## 3         mon      226        1   999        0 nonexistent          1.1
## 4         mon      151        1   999        0 nonexistent          1.1
## 5         mon      307        1   999        0 nonexistent          1.1
## 6         mon      198        1   999        0 nonexistent          1.1
##   cons.price.idx cons.conf.idx euribor3m nr.employed  y
## 1         93.994         -36.4     4.857        5191 no
## 2         93.994         -36.4     4.857        5191 no
## 3         93.994         -36.4     4.857        5191 no
## 4         93.994         -36.4     4.857        5191 no
## 5         93.994         -36.4     4.857        5191 no
## 6         93.994         -36.4     4.857        5191 no
```

Structure of the dataset

There are total of 21 columns and 41,118 observations in the dataset. 10 variables are numeric and 11 variables are characters including target variable that is the outcome of the call.

```
str(bank)
```

```
## 'data.frame':    41188 obs. of  21 variables:
##  $ age           : int  56 57 37 40 56 45 59 41 24 25 ...
##  $ job           : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1 8 8 1 2 10 8 ...
##  $ marital       : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...
##  $ education     : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 6 8 6 4 ...
##  $ default       : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...
##  $ housing       : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...
##  $ loan          : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1 1 ...
##  $ contact       : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...
##  $ month         : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...
##  $ day_of_week   : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ duration      : int  261 149 226 151 307 198 139 217 380 50 ...
##  $ campaign      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays         : int  999 999 999 999 999 999 999 999 999 999 ...
##  $ previous      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome      : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ emp.var.rate  : num  1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...
##  $ cons.price.idx: num  94 94 94 94 94 ...
##  $ cons.conf.idx : num  -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...
##  $ euribor3m     : num  4.86 4.86 4.86 4.86 4.86 ...
##  $ nr.employed   : num  5191 5191 5191 5191 5191 ...
##  $ y             : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
```

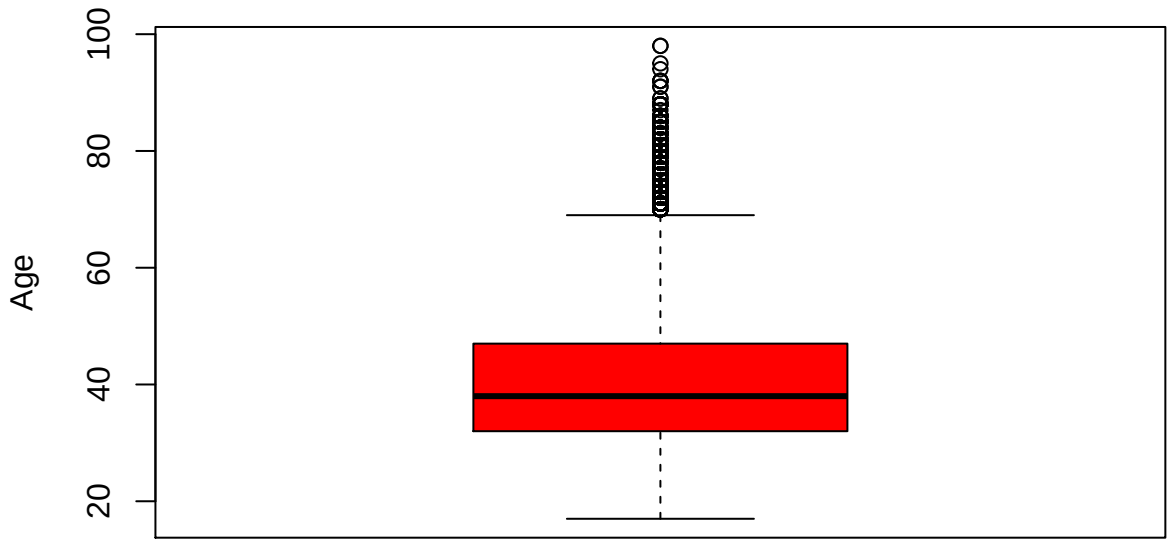
y-It is categorical variable, Yes representing that client has subscribed a term deposit? The data is considered to be imbalanced due a vast difference in class of outcome variable, there are 11% records for yes i.e subscribe for term deposit and 89% for not interested customers.

```
table (bank$y)
```

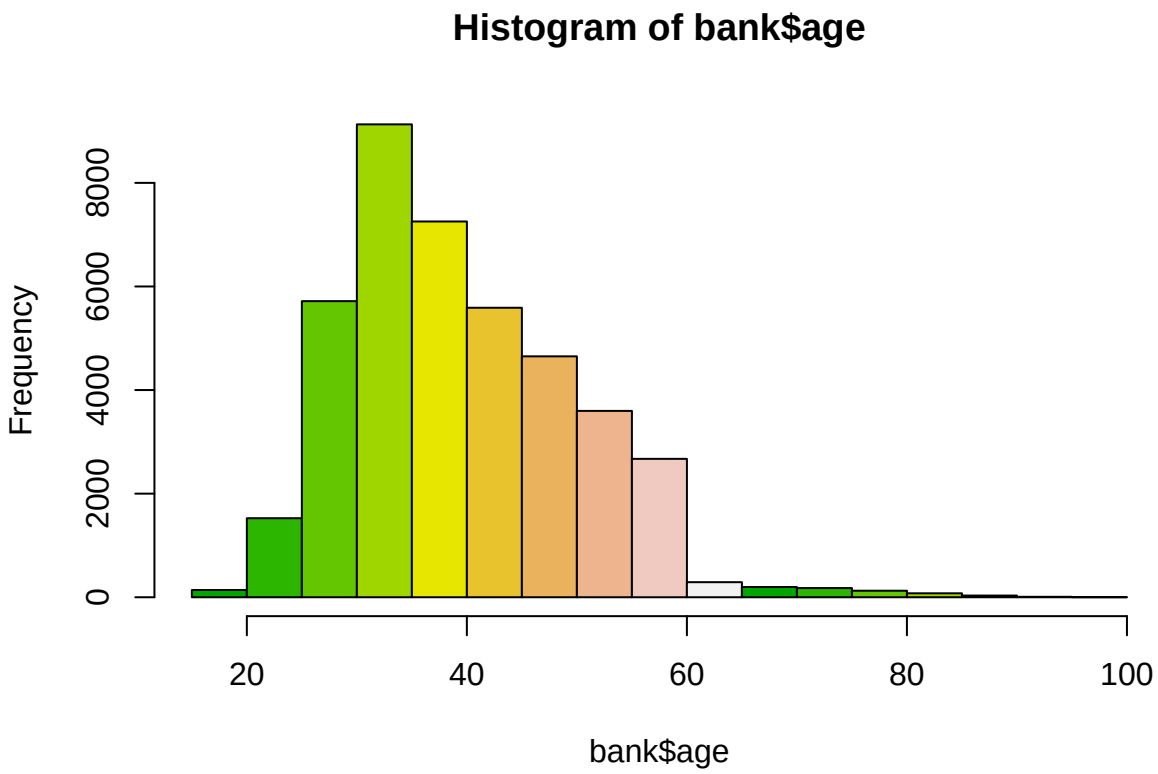
```
##
##    no   yes
## 36548  4640
```

There are no such outliers in age variable. Majority of the records have age 60 or below.

```
boxplot(bank$age, xlab="", ylab="Age",vertical=TRUE,col=2)
```

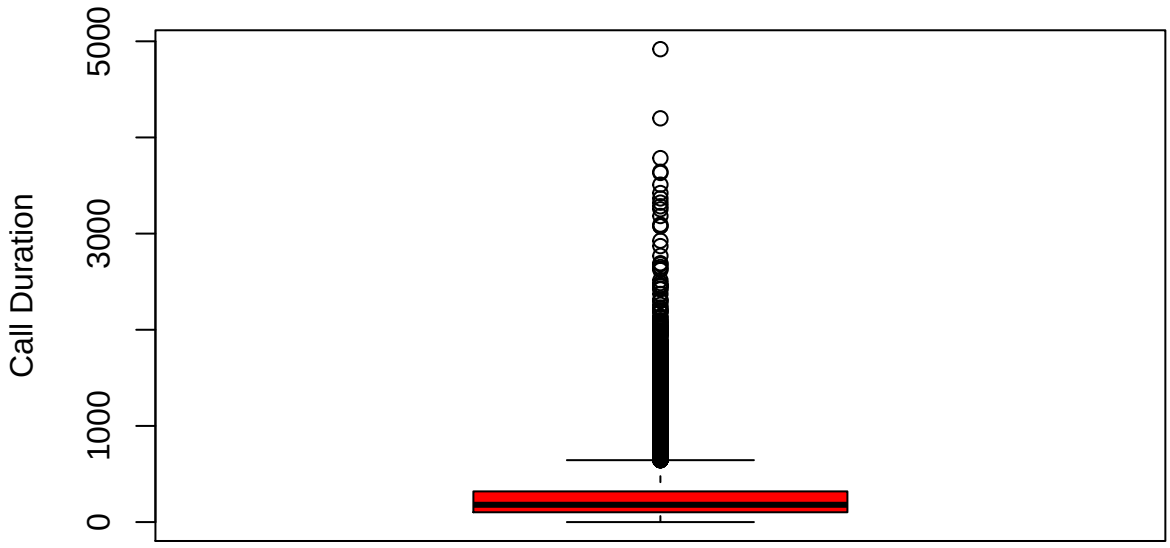



```
hist(bank$age,col=terrain.colors(10))
```

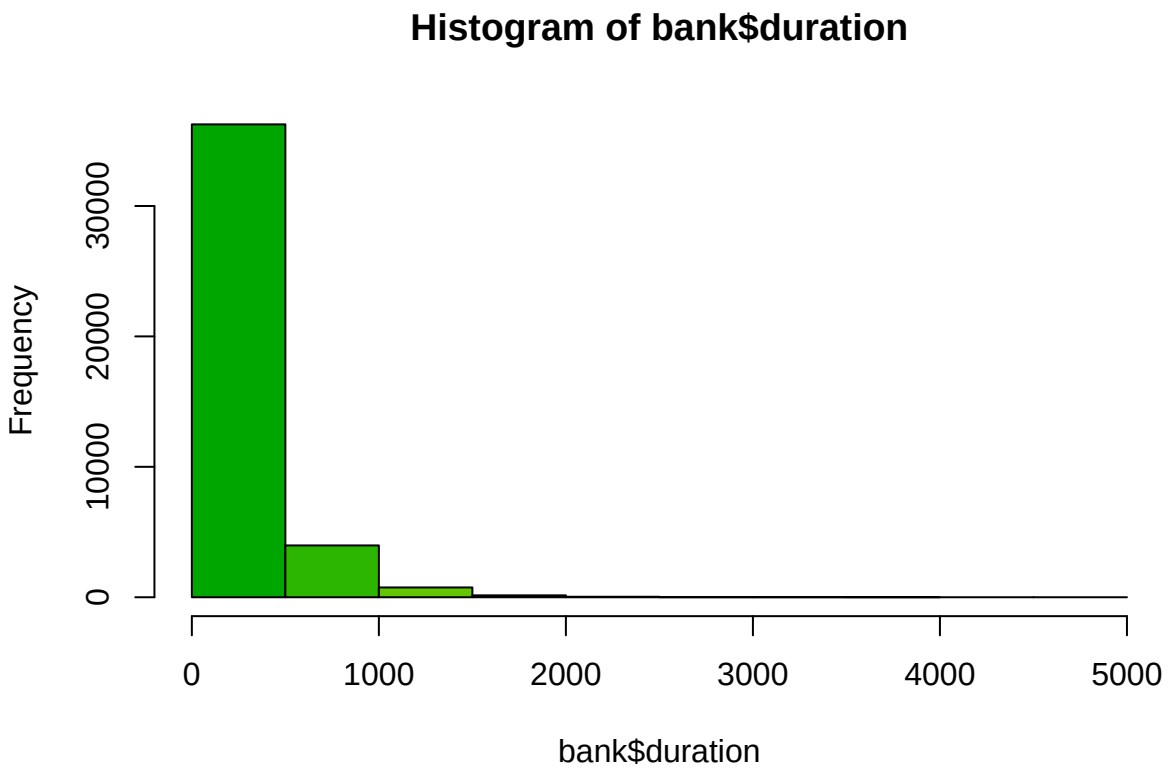

Histogram of bank$age
Frequency
0
2000
4000
6000
8000
20
40
60
80
100
bank$age

Duration represents the duration of last contact to the customer. Variable is highly correlated with the outcome variable.

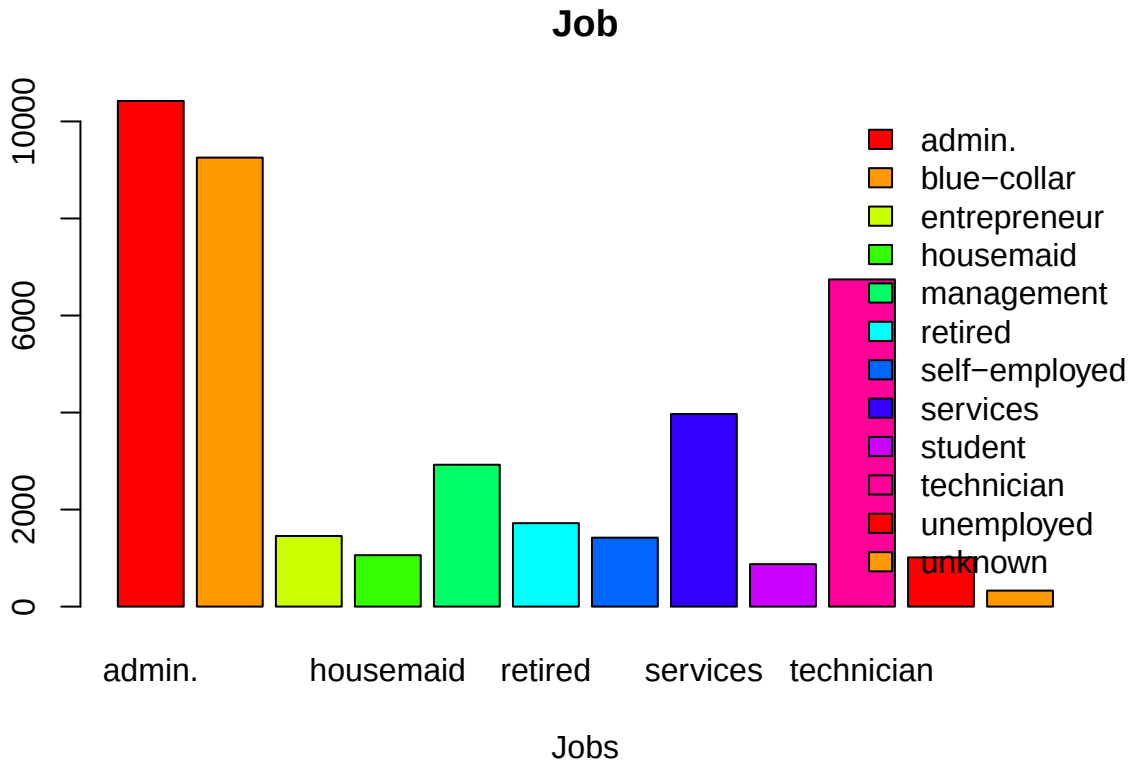
```
boxplot(bank$duration, xlab="", ylab="Call Duration",vertical=TRUE,col=2)
```




```
hist(bank$duration,col=terrain.colors(10))
```


Histogram of bank$duration
Frequency
0
10000
20000
30000
0
1000
2000
3000
4000
5000
bank$duration

```
barplot(table(bank$job), main="Job", xlab="Jobs",col=rainbow(10),legend.text = TRUE, beside=FALSE,args.
```



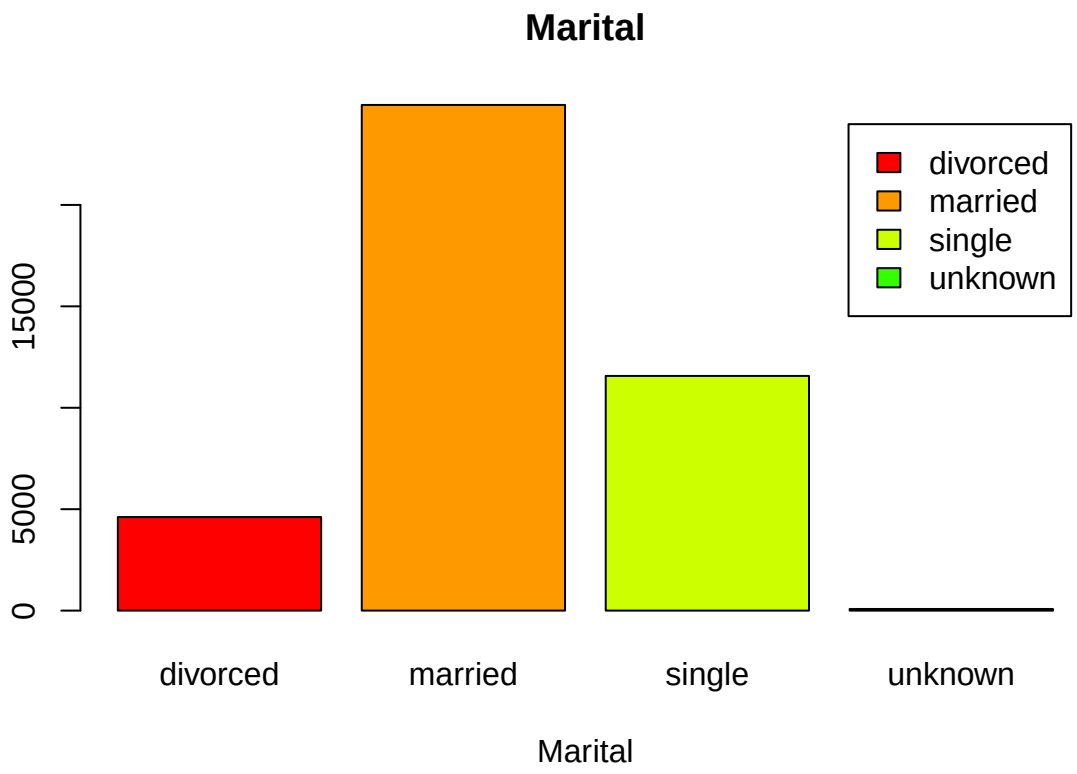

```
table (bank$job, bank$y)
```

```
##
##                   no  yes
##   admin.        9070 1352
##   blue-collar   8616  638
##   entrepreneur  1332  124
##   housemaid      954  106
##   management    2596  328
##   retired       1286  434
##   self-employed 1272  149
##   services      3646  323
##   student        600  275
##   technician    6013  730
##   unemployed     870  144
##   unknown        293   37
```

```
chisq.test(bank$job, bank$y, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$job and bank$y
## X-squared = 961.24, df = 11, p-value < 2.2e-16
```

```
barplot(table(bank$marital), main="Marital", xlab="Marital", col=rainbow(10),legend.text = TRUE, beside=
```




```
table (bank$marital, bank$y)
```

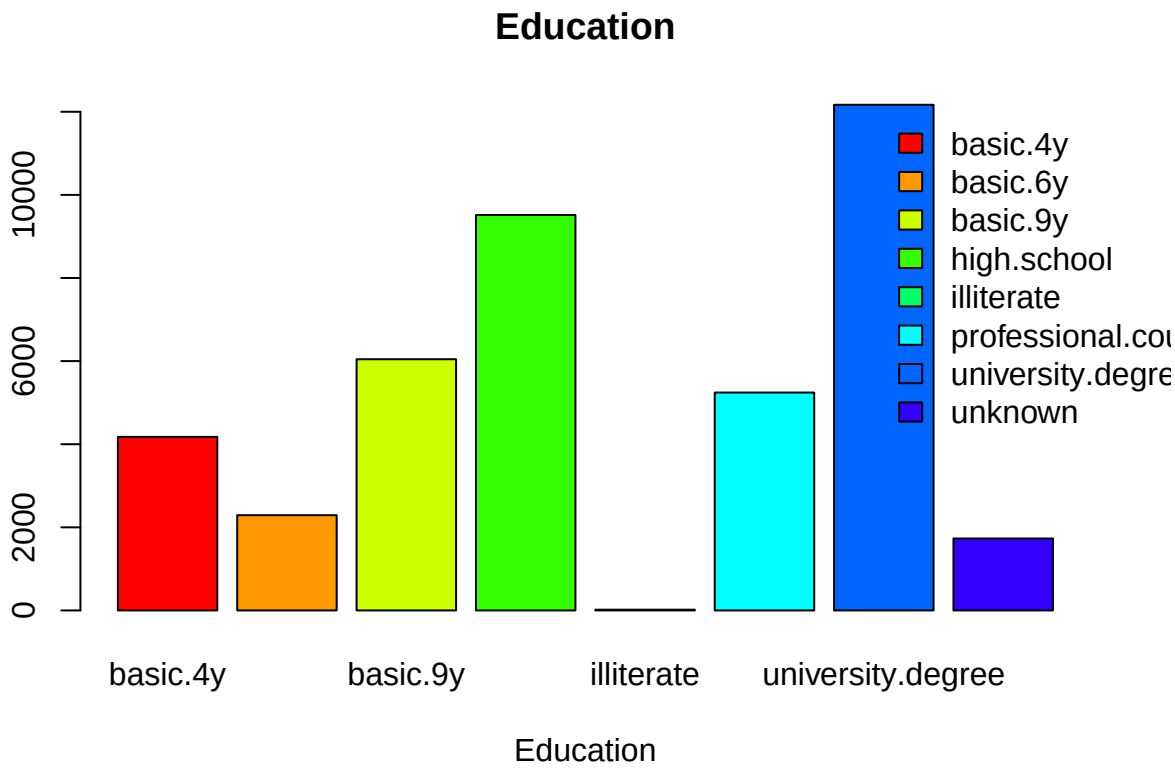
```
##
##               no   yes
##   divorced  4136   476
##   married  22396  2532
##   single    9948  1620
##   unknown     68    12
```

```
chisq.test(bank$marital, bank$y, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$marital and bank$y
## X-squared = 122.66, df = 3, p-value < 2.2e-16
```

In Education variable, class illiterate had very less instances. With the class 'Illiterate' R suggested that the approximation may be incorrect. Therefore the class was removed.

```
barplot(table(bank$education), main="Education", xlab="Education", col=rainbow(10),legend.text = TRUE,,a
```




```
table (bank$education, bank$y)
```

```
##
##                        no   yes
##   basic.4y             3748   428
##   basic.6y             2104   188
##   basic.9y             5572   473
##   high.school          8484  1031
##   illiterate             14     4
##   professional.course  4648   595
##   university.degree   10498  1670
##   unknown              1480   251
```

```
chisq.test(bank$education, bank$y, correct=FALSE)
```

```
## Warning in chisq.test(bank$education, bank$y, correct = FALSE): Chi-squared
## approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
```

```
## data:  bank$education and bank$y
## X-squared = 193.11, df = 7, p-value < 2.2e-16
```

```
bank <- bank %>% filter(education != "illiterate")
chisq.test(bank$education, bank$y, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$education and bank$y
## X-squared = 191.01, df = 6, p-value < 2.2e-16
```

Default variable explains if the client has default on credit products. Class 'yes' has very low records therefore the records will be excluded.

```
table (bank$default, bank$y)
```

```
##
##              no   yes
##   no      28383  4194
##   unknown  8148   442
##   yes         3     0
```

```
chisq.test(bank$default, bank$y, correct=FALSE)
```

```
## Warning in chisq.test(bank$default, bank$y, correct = FALSE): Chi-squared
## approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$default and bank$y
## X-squared = 406.71, df = 2, p-value < 2.2e-16
```

```
bank <- bank %>% filter(default != "yes")
chisq.test(bank$default, bank$y, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$default and bank$y
## X-squared = 406.3, df = 1, p-value < 2.2e-16
```

Housing represents if the customer have any House loan. The Chi Square value 0.05 which can be consider at the border of 95% significance value.

```
table (bank$housing, bank$y)
```

```
##
##              no   yes
##   no      16587  2025
##   unknown   883   107
##   yes     19061  2504
```

```
chisq.test(bank$housing, bank$y, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$housing and bank$y
## X-squared = 5.5553, df = 2, p-value = 0.06218
```

Loan represents if the customer have any personal loan.

The Chi Square value 0.57 which shows the loan doesnt have a signifance on the outcome variable Y. Therefore, we can consider after initial results to remove this variable.

```
table (bank$loan, bank$y)
```

```
##
##               no   yes
##   no      30085  3847
##   unknown   883   107
##   yes      5563   682
```

```
chisq.test(bank$loan, bank$y, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$loan and bank$y
## X-squared = 1.1248, df = 2, p-value = 0.5698
```

```
table (bank$contact, bank$y)
```

```
##
##                 no   yes
##   cellular   22276  3850
##   telephone 14255   786
```

```
chisq.test(bank$contact, bank$y, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$contact and bank$y
## X-squared = 863.98, df = 1, p-value < 2.2e-16
```

```
table (bank$day_of_week, bank$y)
```

```
##
##          no  yes
##   fri 6977  846
##   mon 7666  847
##   thu 7574 1043
##   tue 7130  952
##   wed 7184  948
```

```
chisq.test(bank$day_of_week, bank$y, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$day_of_week and bank$y
## X-squared = 25.795, df = 4, p-value = 3.48e-05
```

```r
table (bank$poutcome, bank$y)
```

```
##
##                   no   yes
##   failure      3645   605
##   nonexistent 32407  3138
##   success       479   893
```

```r
chisq.test(bank$poutcome, bank$y, correct=FALSE)
```

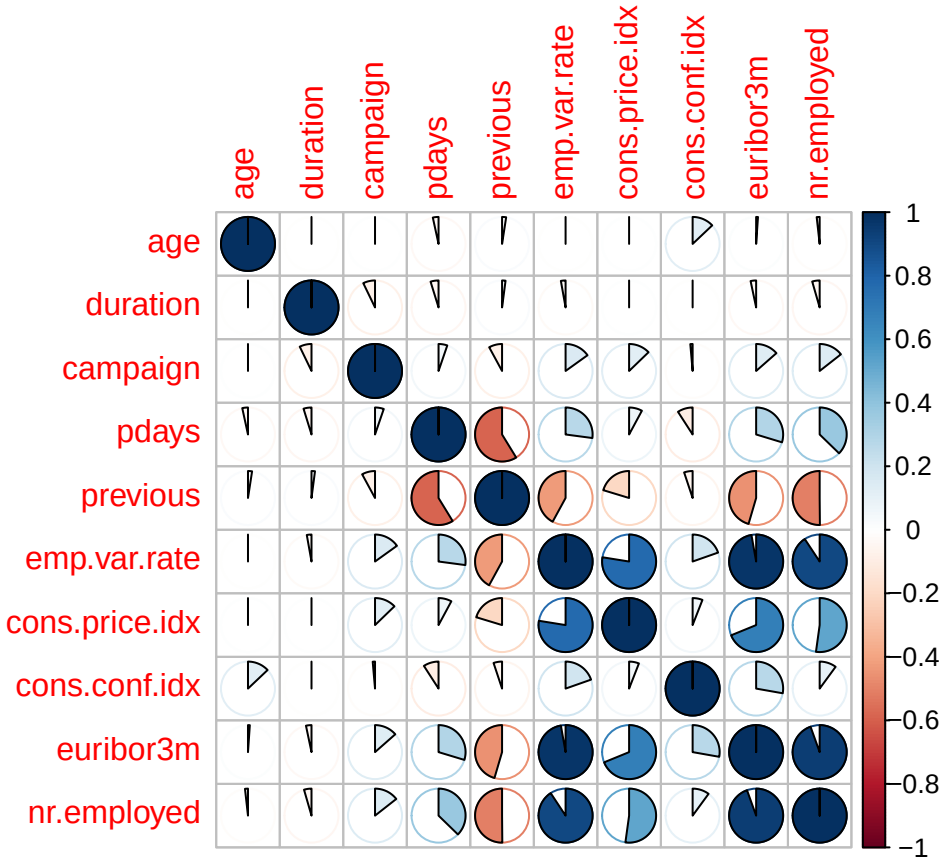
```
##
##  Pearson's Chi-squared test
##
## data:  bank$poutcome and bank$y
## X-squared = 4225.9, df = 2, p-value < 2.2e-16
```

Checking Correlation for all the numeric variables. Strong correlation is observed for all economic variables emp.var.rate, cons.price.idx, cons.conf.idx, euribor3m, and nr.employed.

```
round(cor(bank[,c(1,11:14,16:20)]),2)
```

```
##                  age duration campaign pdays previous emp.var.rate
## age             1.00     0.00     0.00 -0.03     0.02         0.00
## duration        0.00     1.00    -0.07 -0.05     0.02        -0.03
## campaign        0.00    -0.07     1.00  0.05    -0.08         0.15
## pdays          -0.03    -0.05     0.05  1.00    -0.59         0.27
## previous        0.02     0.02    -0.08 -0.59     1.00        -0.42
## emp.var.rate    0.00    -0.03     0.15  0.27    -0.42         1.00
## cons.price.idx  0.00     0.01     0.13  0.08    -0.20         0.78
## cons.conf.idx   0.13    -0.01    -0.01 -0.09    -0.05         0.20
## euribor3m       0.01    -0.03     0.14  0.30    -0.45         0.97
## nr.employed    -0.02    -0.04     0.14  0.37    -0.50         0.91
##                cons.price.idx cons.conf.idx euribor3m nr.employed
## age                      0.00          0.13      0.01       -0.02
## duration                 0.01         -0.01     -0.03       -0.04
## campaign                 0.13         -0.01      0.14        0.14
## pdays                    0.08         -0.09      0.30        0.37
## previous                -0.20         -0.05     -0.45       -0.50
## emp.var.rate             0.78          0.20      0.97        0.91
## cons.price.idx           1.00          0.06      0.69        0.52
## cons.conf.idx            0.06          1.00      0.28        0.10
## euribor3m                0.69          0.28      1.00        0.95
## nr.employed              0.52          0.10      0.95        1.00
```

```
corrplot(cor(bank[,c(1,11:14,16:20)]), method = "pie")
```


age
duration
campaign
pdays
previous
emp.var.rate
cons.price.idx
cons.conf.idx
euribor3m
nr.employed
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1

To avoid multicolinearity, Variance Inflation Factor was checked for different group of variables Social & Economic and Campaign. Value of VIF seems to be really high therefore i have considered to remove the economic variables.

For campaign related variables, duration is highly correlated to outcome variable but this could only be known when we make the call, so i will exclude from the analysis.

pdays and previous are the variables related previous contact. pdays have high number of no contact value '999' therefore i will remove pdays from the model.

Used library "faraway" to use the function of "vif"

```
mymodel_eco <- glm(y ~ emp.var.rate + cons.price.idx + cons.conf.idx + euribor3m + nr.employed ,data=ba

mymodel_cam <- glm(y ~ duration + pdays + previous,data=bank, family=binomial)

summary(mymodel_eco)
```

```
##
## Call:
## glm(formula = y ~ emp.var.rate + cons.price.idx + cons.conf.idx +
##     euribor3m + nr.employed, family = binomial, data = bank)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1759  -0.3703  -0.3388  -0.2658   2.6193
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -21.784535  13.084147  -1.665   0.0959 .
## emp.var.rate    -0.490054   0.057568  -8.513  < 2e-16 ***
## cons.price.idx   0.628745   0.083349   7.543 4.58e-14 ***
## cons.conf.idx    0.034177   0.005016   6.814 9.50e-12 ***
## euribor3m        0.053554   0.072358   0.740   0.4592
## nr.employed     -0.007404   0.001218  -6.079 1.21e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28977  on 41166  degrees of freedom
## Residual deviance: 24329  on 41161  degrees of freedom
## AIC: 24341
##
## Number of Fisher Scoring iterations: 5
```

```
summary(mymodel_cam)
```

```
##
## Call:
## glm(formula = y ~ duration + pdays + previous, family = binomial,
##     data = bank)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -5.5561  -0.3778  -0.2987  -0.2555   2.6781
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.178e+00  8.390e-02  -14.05   <2e-16 ***
## duration     3.891e-03  6.206e-05   62.70   <2e-16 ***
## pdays       -2.526e-03  8.034e-05  -31.44   <2e-16 ***
## previous     4.053e-01  3.710e-02   10.93   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28977  on 41166  degrees of freedom
## Residual deviance: 21345  on 41163  degrees of freedom
## AIC: 21353
##
## Number of Fisher Scoring iterations: 5
```

```
vif(mymodel_eco)
```

```
##   emp.var.rate cons.price.idx  cons.conf.idx      euribor3m    nr.employed
##      336.68210       95.81285       22.18179      648.41374      318.87454
```

```
table(bank$pdays,bank$y)
```

```
##
##          no   yes
##   0       5    10
##   1      18     8
##   2      24    37
##   3     141   298
##   4      55    63
##   5      17    29
##   6     123   288
##   7      20    40
##   8       6    12
##   9      29    35
##   10     22    30
##   11     13    15
##   12     32    26
##   13      8    28
##   14      9    11
##   15      8    16
##   16      5     6
##   17      6     2
##   18      3     4
##   19      2     1
##   20      1     0
##   21      0     2
##   22      1     2
##   25      0     1
##   26      0     1
##   27      0     1
```

```
##   999 35983  3670
```

```
vif(mymodel_cam)
```

```
##  duration     pdays  previous
## 10.659277  9.281471 13.879066
```

Converting categorical variable to numeric, by creating dummy variables using one hot encoding.

```
bank$job_1 <- as.numeric(bank$job == "admin")
bank$job_2 <- as.numeric(bank$job == "blue_collar")
bank$job_3 <- as.numeric(bank$job == "entrepreneur")
bank$job_4 <- as.numeric(bank$job == "housemaid")
bank$job_5 <- as.numeric(bank$job == "management")
bank$job_6 <- as.numeric(bank$job == "retired")
bank$job_7 <- as.numeric(bank$job == "self-employed")
bank$job_8 <- as.numeric(bank$job == "services")
bank$job_9 <- as.numeric(bank$job == "student")
bank$job_10 <- as.numeric(bank$job == "technician")
bank$job_11 <- as.numeric(bank$job == "unemployed")
bank$job_12 <- as.numeric(bank$job == "unknown")

for(LEVEL in unique(bank$marital)){
  bank[paste("marital", LEVEL, sep = "_")] <- ifelse(bank$marital== LEVEL, 1, 0)}

for(LEVEL in unique(bank$education)){
  bank[paste("education", LEVEL, sep = "_")] <- ifelse(bank$education == LEVEL, 1, 0)}

for(LEVEL in unique(bank$default)){
  bank[paste("default", LEVEL, sep = "_")] <- ifelse(bank$default == LEVEL, 1, 0)}

for(LEVEL in unique(bank$housing)){
  bank[paste("housing", LEVEL, sep = "_")] <- ifelse(bank$housing == LEVEL, 1, 0)}

for(LEVEL in unique(bank$loan)){
  bank[paste("loan", LEVEL, sep = "_")] <- ifelse(bank$loan == LEVEL, 1, 0)}

for(LEVEL in unique(bank$contact)){
  bank[paste("contact", LEVEL, sep = "_")] <- ifelse(bank$contact == LEVEL, 1, 0)}

for(LEVEL in unique(bank$month)){
  bank[paste("month", LEVEL, sep = "_")] <- ifelse(bank$month == LEVEL, 1, 0)}

for(LEVEL in unique(bank$day_of_week)){
  bank[paste("day_of_week", LEVEL, sep = "_")] <- ifelse(bank$day_of_week == LEVEL, 1, 0)}

for(LEVEL in unique(bank$poutcome)){
  bank[paste("poutcome", LEVEL, sep = "_")] <- ifelse(bank$poutcome == LEVEL, 1, 0)}
```

Remove all the original categorical variables for which the dummy variables are created. Also removing the social & economic variables, duration and pdays.

```
bank$job <- NULL
bank$marital <- NULL
bank$education <- NULL
bank$default <- NULL
bank$housing <- NULL
bank$loan <- NULL #we will consider to remove this variable after initial
bank$contact <- NULL
bank$month <- NULL
bank$day_of_week <- NULL
bank$poutcome <- NULL

bank$duration <- NULL
bank$pdays <- NULL

bank$emp.var.rate <- NULL
bank$cons.price.idx <- NULL
bank$cons.conf.idx <- NULL
bank$euribor3m <- NULL
bank$nr.employed <- NULL
```

```
colnames(bank)
```

```
##  [1] "age"                            "campaign"
##  [3] "previous"                       "y"
##  [5] "job_1"                          "job_2"
##  [7] "job_3"                          "job_4"
##  [9] "job_5"                          "job_6"
## [11] "job_7"                          "job_8"
## [13] "job_9"                          "job_10"
## [15] "job_11"                         "job_12"
## [17] "marital_married"                "marital_single"
## [19] "marital_divorced"               "marital_unknown"
## [21] "education_basic.4y"             "education_high.school"
## [23] "education_basic.6y"             "education_basic.9y"
## [25] "education_professional.course" "education_unknown"
## [27] "education_university.degree"    "default_no"
## [29] "default_unknown"                "housing_no"
## [31] "housing_yes"                    "housing_unknown"
## [33] "loan_no"                        "loan_yes"
## [35] "loan_unknown"                   "contact_telephone"
## [37] "contact_cellular"               "month_may"
## [39] "month_jun"                      "month_jul"
## [41] "month_aug"                      "month_oct"
## [43] "month_nov"                      "month_dec"
## [45] "month_mar"                      "month_apr"
## [47] "month_sep"                      "day_of_week_mon"
## [49] "day_of_week_tue"                "day_of_week_wed"
## [51] "day_of_week_thu"                "day_of_week_fri"
## [53] "poutcome_nonexistent"           "poutcome_failure"
```

```
## [55] "poutcome_success"
```

```
dim(bank)
```

```
## [1] 41167    55
```

Rearranging the variable to have outcome First and then all other variables.

```
bank <- bank[,c(4,1,2,3,5:54)]
```

Splitting the dataset Bank into Training and Test set by the ratio of 80% and 20% respectively.

```
set.seed(123)

split <- sample.split(bank$y, SplitRatio = 0.80)

train <- subset(bank, split==TRUE)
test <- subset(bank, split==FALSE)

table(train$y)
```

```
##
##    no   yes
## 29225  3709
```

```
table(test$y)
```

```
##
##   no  yes
## 7306  927
```

FITTING LOGISTIC REGRESSION MODEL TO THE TRAINING DATASET

```
model_LR <- glm(formula = y ~ ., data=train, family=binomial)
summary(model_LR)
```

```
##
## Call:
## glm(formula = y ~ ., family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4158  -0.4642  -0.3767  -0.2704   3.1196
##
## Coefficients: (11 not defined because of singularities)
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   0.412807   0.446343   0.925 0.355036
## age                           0.005652   0.002316   2.440 0.014687 *
```

```
## campaign                       -0.074732   0.010383  -7.198 6.12e-13 ***
## previous                        0.337484   0.060284   5.598 2.17e-08 ***
## job_1                                NA         NA      NA       NA
## job_2                                NA         NA      NA       NA
## job_3                          -0.119575   0.114854  -1.041 0.297825
## job_4                          -0.011026   0.135075  -0.082 0.934941
## job_5                          -0.105851   0.080132  -1.321 0.186516
## job_6                           0.555919   0.096060   5.787 7.16e-09 ***
## job_7                          -0.026746   0.107764  -0.248 0.803985
## job_8                          -0.142284   0.077825  -1.828 0.067512 .
## job_9                           0.752213   0.105589   7.124 1.05e-12 ***
## job_10                         -0.094131   0.062924  -1.496 0.134667
## job_11                          0.238330   0.118008   2.020 0.043423 *
## job_12                         -0.005288   0.231282  -0.023 0.981757
## marital_married                -0.281878   0.377578  -0.747 0.455340
## marital_single                 -0.147744   0.378378  -0.390 0.696191
## marital_divorced               -0.405028   0.381638  -1.061 0.288558
## marital_unknown                      NA         NA      NA       NA
## education_basic.4y             -0.264871   0.079508  -3.331 0.000864 ***
## education_high.school          -0.107286   0.056350  -1.904 0.056921 .
## education_basic.6y             -0.201857   0.100214  -2.014 0.043981 *
## education_basic.9y             -0.314859   0.068284  -4.611 4.01e-06 ***
## education_professional.course -0.059819   0.069271  -0.864 0.387837
## education_unknown               0.057751   0.097201   0.594 0.552421
## education_university.degree          NA         NA      NA       NA
## default_no                      0.612112   0.063284   9.672  < 2e-16 ***
## default_unknown                      NA         NA      NA       NA
## housing_no                      0.027358   0.138624   0.197 0.843551
## housing_yes                     0.003309   0.137475   0.024 0.980799
## housing_unknown                      NA         NA      NA       NA
## loan_no                         0.023634   0.053865   0.439 0.660835
## loan_yes                             NA         NA      NA       NA
## loan_unknown                         NA         NA      NA       NA
## contact_telephone              -0.959263   0.057547 -16.669  < 2e-16 ***
## contact_cellular                     NA         NA      NA       NA
## month_may                      -1.291123   0.119010 -10.849  < 2e-16 ***
## month_jun                      -0.553738   0.126764  -4.368 1.25e-05 ***
## month_jul                      -1.250252   0.122289 -10.224  < 2e-16 ***
## month_aug                      -1.312396   0.121254 -10.824  < 2e-16 ***
## month_oct                       0.242659   0.143924   1.686 0.091791 .
## month_nov                      -1.393260   0.125854 -11.070  < 2e-16 ***
## month_dec                       0.522295   0.215952   2.419 0.015582 *
## month_mar                       0.550497   0.151347   3.637 0.000275 ***
## month_apr                      -0.553674   0.124690  -4.440 8.98e-06 ***
## month_sep                            NA         NA      NA       NA
## day_of_week_mon                -0.135594   0.063045  -2.151 0.031497 *
## day_of_week_tue                 0.115981   0.061869   1.875 0.060844 .
## day_of_week_wed                 0.132395   0.062150   2.130 0.033150 *
## day_of_week_thu                 0.082966   0.060700   1.367 0.171683
## day_of_week_fri                      NA         NA      NA       NA
## poutcome_nonexistent           -1.671593   0.114547 -14.593  < 2e-16 ***
## poutcome_failure               -2.098623   0.087416 -24.007  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 23183  on 32933  degrees of freedom
## Residual deviance: 19378  on 32891  degrees of freedom
## AIC: 19464
##
## Number of Fisher Scoring iterations: 6
```

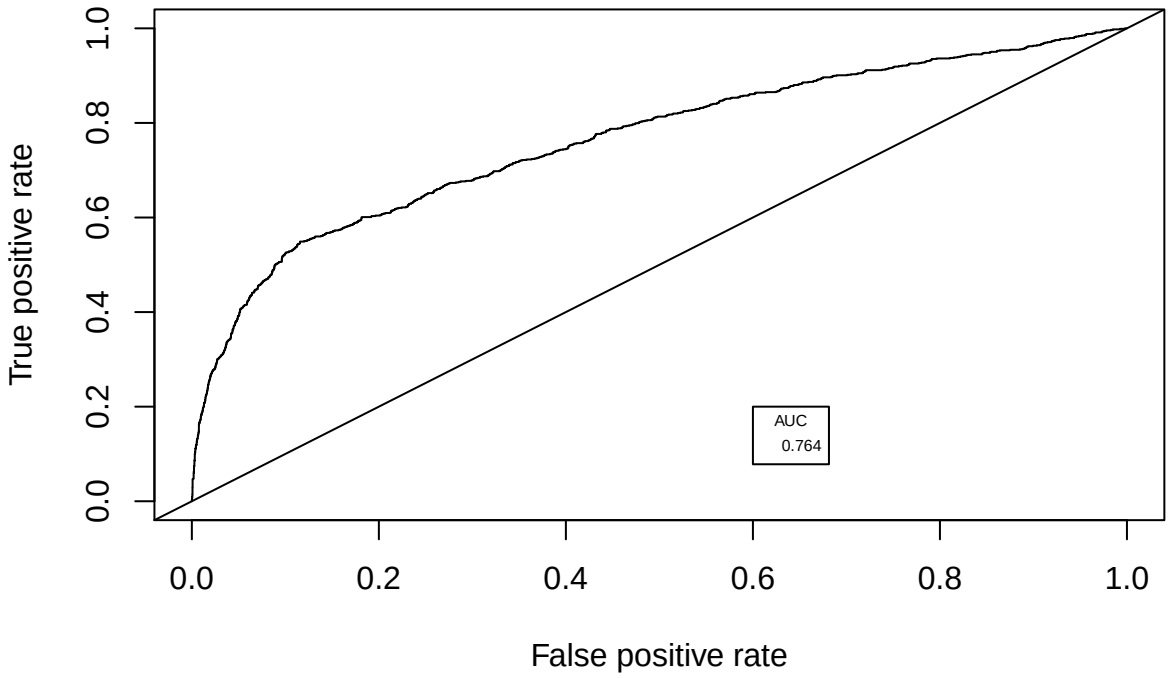
```
prob_pred = predict(model_LR, type='response', newdata=test[-1])
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
LR_pred = ifelse(prob_pred > 0.5, 1,0)

LR_CM = table(test[,1],LR_pred)
LR_CM
```

```
##      LR_pred
##          0    1
##   no  7222   84
##   yes  749  178
```

```
#Accuracy is calculated at 89%

pred<-prediction(prob_pred, test$y)
eval <- performance(pred,"tpr","fpr")
plot(eval, colorize=F)
abline(a=0, b=1)

auc <- performance(pred,"auc")
auc <- unlist(slot(auc,"y.values"))
auc <- round(auc,4)
legend(.6,.2,auc, title="AUC", cex=0.5)
```



FITTING RANDOM FOREST MODEL TO THE TRAINING DATASET

```
model_rf <- randomForest(y~.,data=train)

library(e1071)

#Model accuracy is at 89% however the sensitivity is on the lower side.

confusionMatrix(predict(model_rf, test), test$y, positive='yes')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   no  yes
##        no  7182  713
##        yes  124  214
##
##                Accuracy : 0.8983
##                  95% CI : (0.8916, 0.9048)
##     No Information Rate : 0.8874
##     P-Value [Acc > NIR] : 0.0007843
##
##                   Kappa : 0.296
```

```
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.23085
##             Specificity : 0.98303
##          Pos Pred Value : 0.63314
##          Neg Pred Value : 0.90969
##              Prevalence : 0.11260
##          Detection Rate : 0.02599
##    Detection Prevalence : 0.04105
##       Balanced Accuracy : 0.60694
##
##        'Positive' Class : yes
##
```