**Output variable -> y**

**y -> Whether the client has subscribed a term deposit or not**

**Binomial ("yes" or "no")**

**Attribute information For bank dataset**

**Input variables:**

**# bank client data:**

**1 - age (numeric)**

**2 - job : type of job (categorical: "admin.","unknown","unemployed","management","housemaid","entrepreneur","student",**

**"blue-collar","self-employed","retired","technician","services")**

**3 - marital : marital status (categorical: "married","divorced","single"; note: "divorced" means divorced or widowed)**

**4 - education (categorical: "unknown","secondary","primary","tertiary")**

**5 - default: has credit in default? (binary: "yes","no")**

**6 - balance: average yearly balance, in euros (numeric)**

**7 - housing: has housing loan? (binary: "yes","no")**

**8 - loan: has personal loan? (binary: "yes","no")**

**# related with the last contact of the current campaign:**

**9 - contact: contact communication type (categorical: "unknown","telephone","cellular")**

**10 - day: last contact day of the month (numeric)**

**11 - month: last contact month of year (categorical: "jan", "feb", "mar", ..., "nov", "dec")**

**12 - duration: last contact duration, in seconds (numeric)**

**# other attributes:**

**13 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)**

**14 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)**

**15 - previous: number of contacts performed before this campaign and for this client (numeric)**

**16 - poutcome: outcome of the previous marketing campaign (categorical: "unknown","other","failure","success")**

**Output variable (desired target):**

**17 - y - has the client subscribed a term deposit? (binary: "yes","no")**

**8. Missing Attribute Values: None**

install.packages("plyr")

install.packages("ggplot2")

**#importing csv files**

View(bank\_data)

**#display first top records from a data set**

head(bank\_data)

age job marital education default balance housing loan contact day month duration

1 58 management married tertiary no 2143 yes no unknown 5 may 261

2 44 technician single secondary no 29 yes no unknown 5 may 151

3 33 entrepreneur married secondary no 2 yes yes unknown 5 may 76

4 47 blue-collar married unknown no 1506 yes no unknown 5 may 92

5 33 unknown single unknown no 1 no no unknown 5 may 198

6 35 management married tertiary no 231 yes no unknown 5 may 139

campaign pdays previous poutcome y

1 1 -1 0 unknown no

2 1 -1 0 unknown no

3 1 -1 0 unknown no

4 1 -1 0 unknown no

5 1 -1 0 unknown no

6 1 -1 0 unknown no

**#display structure of the data set**

str(bank\_data)

'data.frame': 45211 obs. of 17 variables:

$ age : int 58 44 33 47 33 35 28 42 58 43 ...

$ job : chr "management" "technician" "entrepreneur" "blue-collar" ...

$ marital : chr "married" "single" "married" "married" ...

$ education: chr "tertiary" "secondary" "secondary" "unknown" ...

$ default : chr "no" "no" "no" "no" ...

$ balance : int 2143 29 2 1506 1 231 447 2 121 593 ...

$ housing : chr "yes" "yes" "yes" "yes" ...

$ loan : chr "no" "no" "yes" "no" ...

$ contact : chr "unknown" "unknown" "unknown" "unknown" ...

$ day : int 5 5 5 5 5 5 5 5 5 5 ...

$ month : chr "may" "may" "may" "may" ...

$ duration : int 261 151 76 92 198 139 217 380 50 55 ...

$ campaign : int 1 1 1 1 1 1 1 1 1 1 ...

$ pdays : int -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...

$ previous : int 0 0 0 0 0 0 0 0 0 0 ...

$ poutcome : chr "unknown" "unknown" "unknown" "unknown" ...

$ y : chr "no" "no" "no" "no" ...

**#in our data set we have total 45211 obs. of 17 variables:**

**#in our data set many categorical variable,to apply logistic regression algorithm**

**# we would convert the categorical data into numerical format**

**#display the column names**

colnames(bank\_data)

[1] "age" "job" "marital" "education" "default" "balance" "housing"

[8] "loan" "contact" "day" "month" "duration" "campaign" "pdays"

[15] "previous" "poutcome" "y"

**#shows dimension of the data set**

dim(bank\_data)

45211 17

**#shows summary**

summary(bank\_data)

age job marital education default

Min. :18.00 Length:45211 Length:45211 Length:45211 Length:45211

1st Qu.:33.00 Class :character Class :character Class :character Class :character

Median :39.00 Mode :character Mode :character Mode :character Mode :character

Mean :40.94

3rd Qu.:48.00

Max. :95.00

balance housing loan contact day

Min. : -8019 Length:45211 Length:45211 Length:45211 Min. : 1.00

1st Qu.: 72 Class :character Class :character Class :character 1st Qu.: 8.00

Median : 448 Mode :character Mode :character Mode :character Median :16.00

Mean : 1362 Mean :15.81

3rd Qu.: 1428 3rd Qu.:21.00

Max. :102127 Max. :31.00

month duration campaign pdays previous

Length:45211 Min. : 0.0 Min. : 1.000 Min. : -1.0 Min. : 0.0000

Class :character 1st Qu.: 103.0 1st Qu.: 1.000 1st Qu.: -1.0 1st Qu.: 0.0000

Mode :character Median : 180.0 Median : 2.000 Median : -1.0 Median : 0.0000

Mean : 258.2 Mean : 2.764 Mean : 40.2 Mean : 0.5803

3rd Qu.: 319.0 3rd Qu.: 3.000 3rd Qu.: -1.0 3rd Qu.: 0.0000

Max. :4918.0 Max. :63.000 Max. :871.0 Max. :275.0000

poutcome y

Length:45211 Length:45211

Class :character Class :character

Mode :character Mode :character

**#checking the null values present in a data set**

sum(is.na(bank\_data))

0

**#there is no null values in our data set**

**##############VISUALIZATION########################**

**#Here i am visualizing the variable using ggplot**

#install.packages("ggplot2")

library(ggplot2)

**#visualization of age**

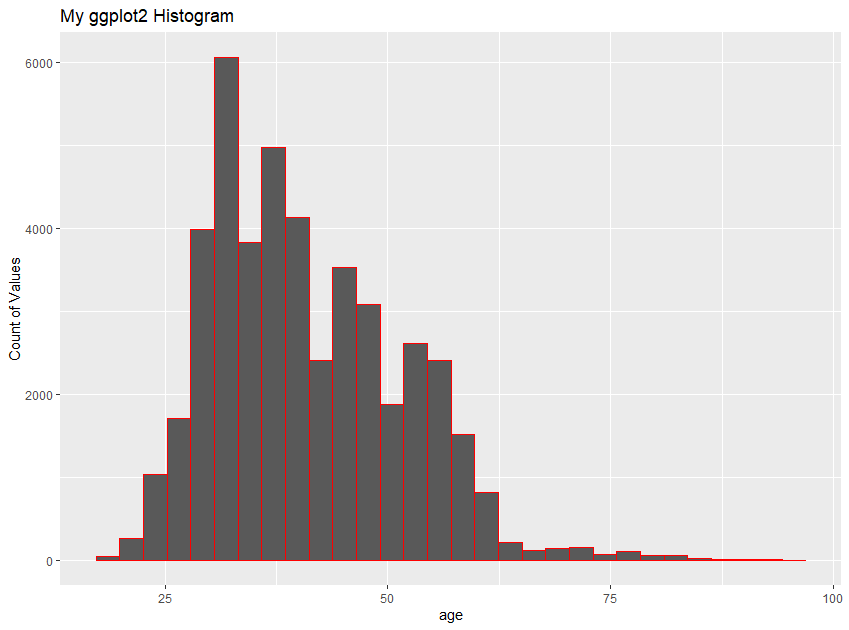
ggplot(bank\_data, aes(x = age)) +

geom\_histogram(col = "red") +

labs(title = "My ggplot2 Histogram",

x = "age",

y = "Count of Values")



**# here we can see that most of the clients age is between 25-50**

**##visualization of average yearly balance, in euros (numeric**)

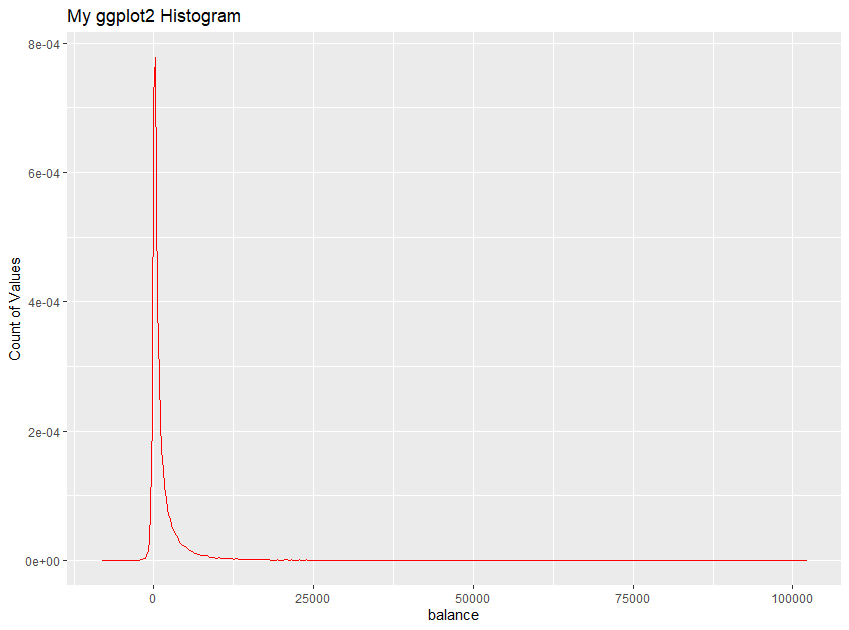
ggplot(bank\_data, aes(x = balance)) +

geom\_density(col = "red") +

labs(title = "My ggplot2 Histogram",

x = "balance",

y = "Count of Values")



**# here we can see that many custemer has negative and zero balance**

**#visualization of duration**

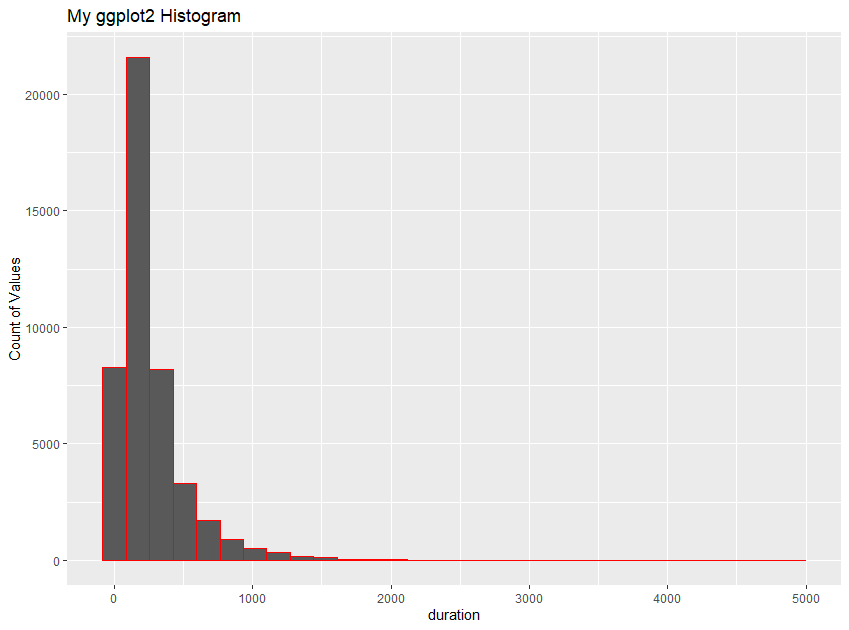
ggplot(bank\_data, aes(x = duration)) +

geom\_histogram(col = "red") +

labs(title = "My ggplot2 Histogram",

x = "duration",

y = "Count of Values")



**#visualization of number of contacts performed during this campaign and for this client**

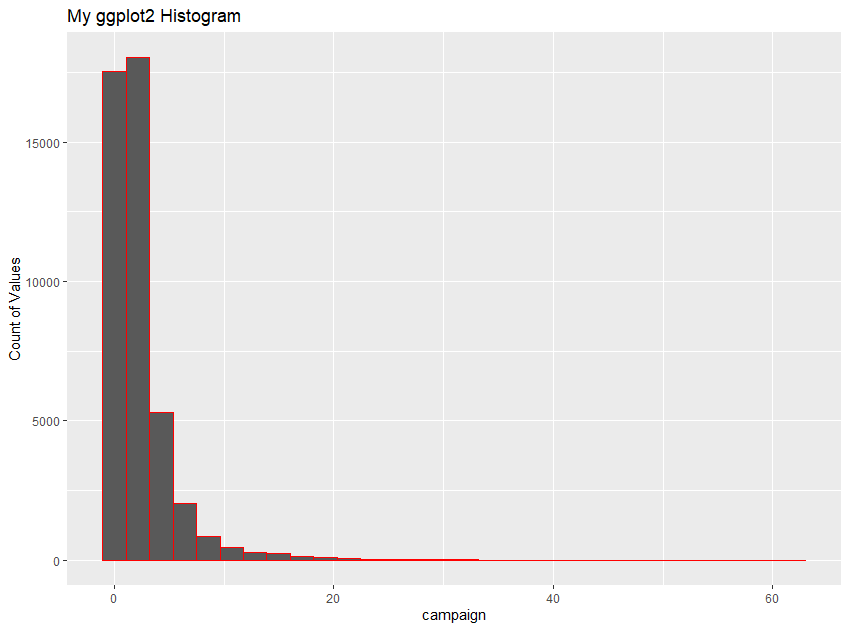
ggplot(bank\_data, aes(x = campaign)) +

geom\_histogram(col = "red") +

labs(title = "My ggplot2 Histogram",

x = "campaign",

y = "Count of Values")



count(bank\_data, "job")

count(bank\_data,"marital")

count(bank\_data,"education")

count(bank\_data,"default")

count(bank\_data,"housing")

count(bank\_data,"loan")

count(bank\_data,"contact")

count(bank\_data,"month")

count(bank\_data,"poutcome")

count(bank\_data,"y")

**#here some variables contain many category, and some variable has less category**

**#so convert categorical data numerical, i am using factor**

bank\_data$job<-as.numeric(factor(bank\_data$job))-1

bank\_data$marital<-as.numeric(factor(bank\_data$marital))-1

bank\_data$education<-as.numeric(factor(bank\_data$education))-1

bank\_data$default<-as.numeric(factor(bank\_data$default))-1

bank\_data$housing<-as.numeric(factor(bank\_data$housing))-1

bank\_data$loan<-as.numeric(factor(bank\_data$loan))-1

bank\_data$contact<-as.numeric(factor(bank\_data$contact))-1

bank\_data$month<-as.numeric(factor(bank\_data$month))-1

bank\_data$poutcome<-as.numeric(factor(bank\_data$poutcome))-1

bank\_data$y<-as.numeric(factor(bank\_data$y))-1

str(bank\_data)

data.frame': 45211 obs. of 17 variables:

$ age : int 58 44 33 47 33 35 28 42 58 43 ...

$ job : num 4 9 2 1 11 4 4 2 5 9 ...

$ marital : num 1 2 1 1 2 1 2 0 1 2 ...

$ education: num 2 1 1 3 3 2 2 2 0 1 ...

$ default : num 0 0 0 0 0 0 0 1 0 0 ...

$ balance : int 2143 29 2 1506 1 231 447 2 121 593 ...

$ housing : num 1 1 1 1 0 1 1 1 1 1 ...

$ loan : num 0 0 1 0 0 0 1 0 0 0 ...

$ contact : num 2 2 2 2 2 2 2 2 2 2 ...

$ day : int 5 5 5 5 5 5 5 5 5 5 ...

$ month : num 8 8 8 8 8 8 8 8 8 8 ...

$ duration : int 261 151 76 92 198 139 217 380 50 55 ...

$ campaign : int 1 1 1 1 1 1 1 1 1 1 ...

$ pdays : int -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...

$ previous : int 0 0 0 0 0 0 0 0 0 0 ...

$ poutcome : num 3 3 3 3 3 3 3 3 3 3 ...

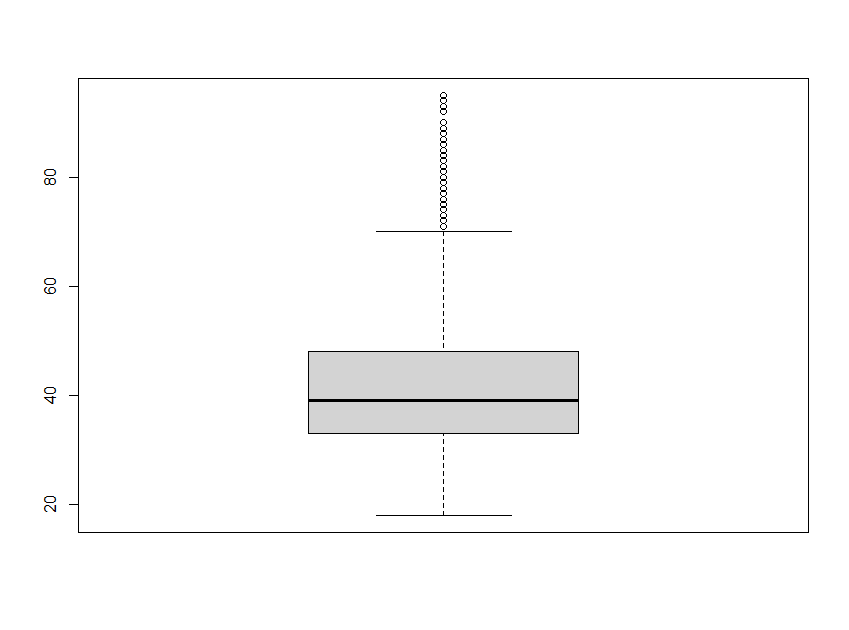
$ y : num 0 0 0 0 0 0 0 0 0 0 ...

**# here we convert all categorical data into a numerical form**

View(bank\_data)

**# Visualization using boxplot**

boxplot(bank\_data$age)



#in box plot representation some variable has outlier

#here age variable has some outliers

#in age column above age above 70 considered as outliers

#so remove all those outliers from the data set

#following code shows the outliers present in the age columns

boxplot(bank\_data$age, plot=FALSE)$out

|  |
| --- |
| [1] 83 75 75 83 75 72 71 76 83 71 85 72 90 71 82 73 74 71 75 78 85 80 71 78 73 75 94 72 83  [30] 79 78 83 73 76 83 77 73 74 73 77 71 71 80 82 75 77 78 78 86 76 72 74 77 74 79 71 74 78  [59] 73 73 95 74 71 78 71 75 75 82 74 81 76 72 75 72 71 72 77 73 85 83 78 73 79 71 71 80 72  [88] 78 79 71 82 81 79 71 79 79 73 75 72 71 75 78 75 80 80 74 73 71 72 72 77 79 73 71 80 71  [117] 83 71 73 78 72 81 81 71 74 75 82 80 76 74 74 77 71 71 77 76 72 74 74 72 73 82 77 71 79  [146] 89 76 81 74 73 71 71 77 72 84 86 72 72 76 73 74 82 71 83 72 73 73 74 72 78 86 76 79 71  [175] 81 77 82 72 73 73 74 71 73 76 71 71 83 71 80 79 77 74 84 95 77 78 79 83 83 73 77 72 74  [204] 83 76 73 81 80 75 74 75 77 83 72 81 73 74 80 72 76 72 77 74 87 76 92 78 82 78 76 73 77  [233] 76 73 82 81 80 71 76 74 75 72 80 84 87 72 76 78 73 78 81 76 71 79 73 72 82 77 72 75 76  [262] 81 77 71 75 74 75 71 78 72 77 83 72 73 81 72 80 80 86 79 71 72 79 87 81 72 75 72 84 78  [291] 89 72 73 85 87 80 81 85 79 76 74 92 77 74 81 71 73 76 80 82 72 77 76 72 78 77 74 76 77  [320] 80 76 76 71 73 84 79 77 75 76 75 71 75 75 77 82 80 72 79 78 77 71 74 71 73 90 73 83 73  [349] 76 71 79 75 80 72 77 74 82 71 75 86 86 72 77 82 72 80 84 86 80 78 73 71 80 75 73 74 76  [378] 76 73 77 80 77 76 79 77 80 74 72 77 83 80 78 78 77 75 72 74 79 73 84 78 71 78 83 80 83  [407] 73 71 76 75 71 75 72 77 74 72 71 80 79 74 72 72 80 79 86 73 93 82 82 77 71 83 80 76 78  [436] 80 93 82 71 79 75 77 77 75 77 75 88 83 72 84 77 80 84 75 83 72 75 74 88 72 74 75 78 81  [465] 82 89 81 72 77 71 79 73 75 86 75 77 77 74 73 78 78 77 71 75 73 71 72 |
|  |
| |  | | --- | | > | |

#removing the outliers

outliers <- boxplot(bank\_data$age, plot=FALSE)$out

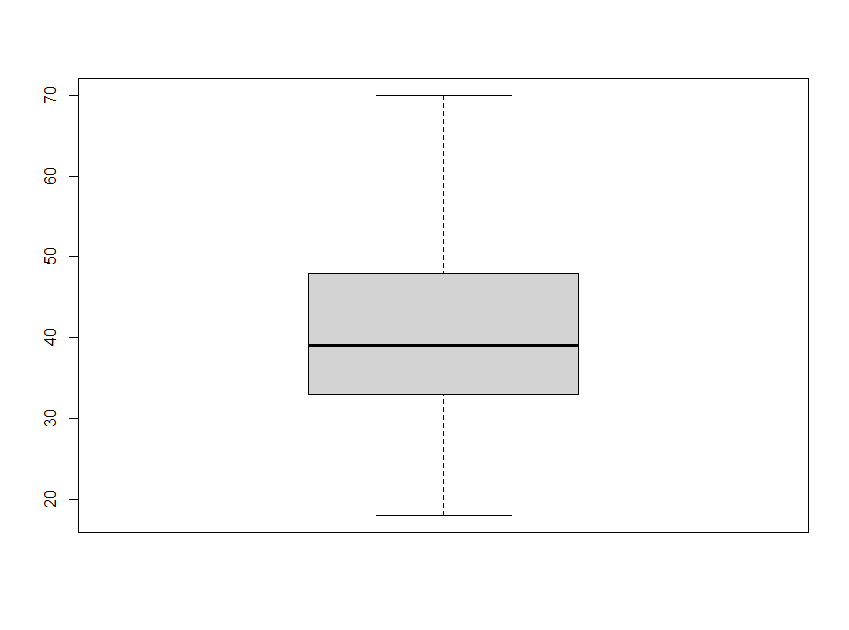
x<- bank\_data

x<- x[-which(x$age %in% outliers),]

bank\_data<-x

**#boxplot of age after removing the outliers**

boxplot(bank\_data$age)



**###SPLITTING THE DATA SET#############3**

**## create train and test data**

install.packages("caTools")

library(caTools)

**##use caTools function to split, SplitRatio for 70%:30% splitting**

data= sample.split(bank\_data,SplitRatio = 0.3)

**## here I am using 70% of data for training and 30% data for testing**

**#subsetting into Test data**

test =subset(bank\_data,data==TRUE)

**#subsetting into Train data**

train=subset(bank\_data,data==FALSE)

**## check number of records present in the data set**

nrow(test)

13154

nrow(train)

31570

head(train)

age job marital education default balance housing loan contact day month duration campaign

1 58 4 1 2 0 2143 1 0 2 5 8 261 1

3 33 2 1 1 0 2 1 1 2 5 8 76 1

4 47 1 1 3 0 1506 1 0 2 5 8 92 1

5 33 11 2 3 0 1 0 0 2 5 8 198 1

6 35 4 1 2 0 231 1 0 2 5 8 139 1

7 28 4 2 2 0 447 1 1 2 5 8 217 1

pdays previous poutcome y

1 -1 0 3 0

3 -1 0 3 0

4 -1 0 3 0

5 -1 0 3 0

6 -1 0 3 0

7 -1 0 3 0

head(test)

age job marital education default balance housing loan contact day month duration campaign

2 44 9 2 1 0 29 1 0 2 5 8 151 1

9 58 5 1 0 0 121 1 0 2 5 8 50 1

10 43 9 2 1 0 593 1 0 2 5 8 55 1

12 29 0 2 1 0 390 1 0 2 5 8 137 1

15 57 7 1 1 0 162 1 0 2 5 8 174 1

19 60 5 1 0 0 60 1 0 2 5 8 219 1

pdays previous poutcome y

2 -1 0 3 0

9 -1 0 3 0

10 -1 0 3 0

12 -1 0 3 0

15 -1 0 3 0

19 -1 0 3 0

####################################################

# BUILD A MODEL###

**#Output variable -> y IS Oor target variable**

**#we have to build a model that predict Whether the client has subscribed a term deposit or not**

**#Binomial ("yes" or "no")**

logistic\_model <- glm(y~.,data=train,family = "binomial")

summary(logistic\_model)

Call:

glm(formula = y ~ ., family = "binomial", data = train)

Deviance Residuals:

Min 1Q Median 3Q Max

-4.5667 -0.4420 -0.2862 -0.1649 3.6035

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -3.586e+00 1.807e-01 -19.847 < 2e-16 \*\*\*

age 7.664e-04 2.235e-03 0.343 0.731634

job 5.189e-03 6.444e-03 0.805 0.420658

marital 1.773e-01 3.775e-02 4.697 2.63e-06 \*\*\*

education 2.097e-01 2.826e-02 7.421 1.16e-13 \*\*\*

default -2.544e-01 1.852e-01 -1.374 0.169426

balance 2.142e-05 5.648e-06 3.793 0.000149 \*\*\*

housing -1.019e+00 4.549e-02 -22.410 < 2e-16 \*\*\*

loan -6.858e-01 6.842e-02 -10.023 < 2e-16 \*\*\*

contact -6.478e-01 3.281e-02 -19.746 < 2e-16 \*\*\*

day -7.257e-03 2.512e-03 -2.889 0.003865 \*\*

month 3.116e-02 6.792e-03 4.587 4.49e-06 \*\*\*

duration 3.938e-03 7.293e-05 53.992 < 2e-16 \*\*\*

campaign -1.288e-01 1.224e-02 -10.520 < 2e-16 \*\*\*

pdays 3.185e-03 3.053e-04 10.434 < 2e-16 \*\*\*

previous 9.699e-02 9.381e-03 10.339 < 2e-16 \*\*\*

poutcome 1.980e-01 3.502e-02 5.655 1.56e-08 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 22312 on 31569 degrees of freedom

Residual deviance: 16553 on 31553 degrees of freedom

AIC: 16587

Number of Fisher Scoring iterations: 6

**##predict the model#######**

probability <- predict(logistic\_model,type=c("response"),train)

View(probability)

head(probability,10)

1 3 4 5 6 7 8

0.029586646 0.005542517 0.018275599 0.086226788 0.017474752 0.014324127 0.028689315

11 13 14

0.016222996 0.062047340 0.017170725

**###create confusion matrix table**

confusion\_matrix<-table(probability>0.5,train$y)

confusion\_matrix

0 1

FALSE 27463 2823

TRUE 530 754

**# here the probability value>0.5, classified as 1, else classified as 0**

**#check the Model Accuracy**

Accuracy<-sum(diag(confusion\_matrix)/sum(confusion\_matrix))

Accuracy

**# we get 89.35% accuracy**

**#check the error rate**

1-Accuracy

**#10.64% error rate**

**########ROC curve##########**

**#ROC curve is a metric describing the trade-off between**

**#the sensitivity (true positive rate, TPR) and specificity (false positive rate, FPR)**

**#of a prediction in all probability cutoffs (thresholds).**

**#It can be used for binary and multiclass classification accuracy checking.**

install.packages("ROCR")

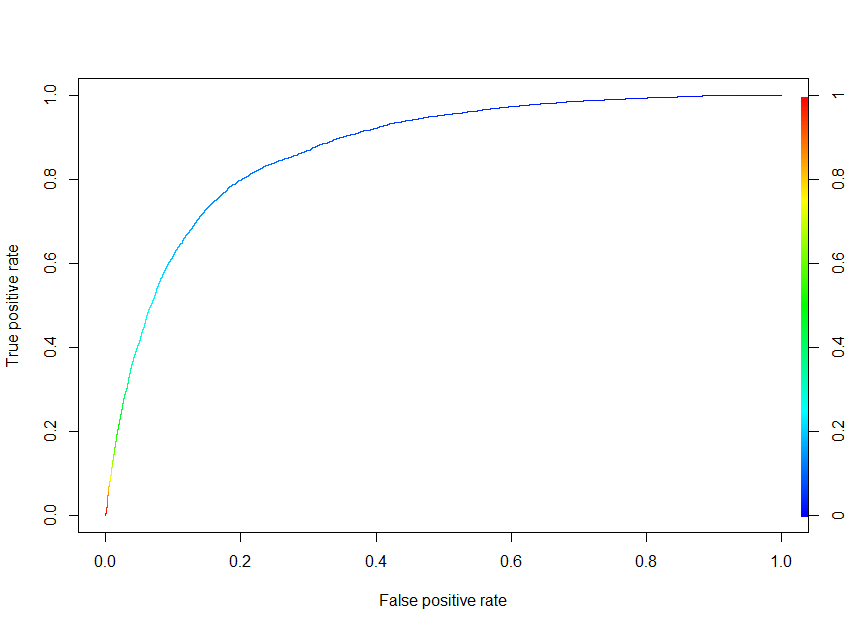
install.packages("ROCR", dependencies = T)

library(ROCR)

roc\_prediction<-prediction(probability,train$y)

roc\_performance<-performance(roc\_prediction,'tpr','fpr')

plot(roc\_performance,colorize=T,text.adj = c(0.5, 0.5))



**# using ROC we can understand how good the model is.**

**# here our model accuracy between 0.9 to 1, so we we can say that our model is outstanding**

###########################################################

**#Predict on Test data set**

pred <- predict(logistic\_model, newdata = test, type = "response")

y\_pred\_num <- ifelse(pred > 0.5, 1, 0)

y\_pred <- factor(y\_pred\_num, levels=c(0, 1))

y\_act<-test$y

**# compare the predicted value in test\_data**

head(test)

age job marital education default balance housing loan contact day month duration campaign

2 44 9 2 1 0 29 1 0 2 5 8 151 1

9 58 5 1 0 0 121 1 0 2 5 8 50 1

10 43 9 2 1 0 593 1 0 2 5 8 55 1

12 29 0 2 1 0 390 1 0 2 5 8 137 1

15 57 7 1 1 0 162 1 0 2 5 8 174 1

19 60 5 1 0 0 60 1 0 2 5 8 219 1

pdays previous poutcome y

2 -1 0 3 0

9 -1 0 3 0

10 -1 0 3 0

12 -1 0 3 0

15 -1 0 3 0

19 -1 0 3 0

test$predict\_data<- y\_pred

head(test)

age job marital education default balance housing loan contact day month duration campaign

2 44 9 2 1 0 29 1 0 2 5 8 151 1

9 58 5 1 0 0 121 1 0 2 5 8 50 1

10 43 9 2 1 0 593 1 0 2 5 8 55 1

12 29 0 2 1 0 390 1 0 2 5 8 137 1

15 57 7 1 1 0 162 1 0 2 5 8 174 1

19 60 5 1 0 0 60 1 0 2 5 8 219 1

pdays previous poutcome y predict\_data

2 -1 0 3 0 0

9 -1 0 3 0 0

10 -1 0 3 0 0

12 -1 0 3 0 0

15 -1 0 3 0 0

19 -1 0 3 0 0

**#compare the accuracy of test data and actual data**

mean(y\_pred== y\_act) **#89.66 %accuracy**

**#create new data frame**

new\_test\_data<-test[,c("y","predict\_data")]

View(new\_test\_data)

|  |
| --- |
| y predict\_data  2 0 0  9 0 0  10 0 0  12 0 0  15 0 0  19 0 0  26 0 0  27 0 0  29 0 0  32 0 0 |
|  |
| |  | | --- | | > | |

**###CONCLUSION#####**

**#Our objective is to create a model for predicting**

**#Whether the client has subscribed a term deposit or not**

**# here we create a good model with accuaracy 89.66%**

**#using this model we have correctly predicted the test data**