**Classify whether application accepted or not using Logistic regression**

**card**

**Factor. Was the application for a credit card accepted?**

**reports**

**Number of major derogatory reports.**

**age**

**Age in years plus twelfths of a year.**

**income**

**Yearly income (in USD 10,000).**

**share**

**Ratio of monthly credit card expenditure to yearly income.**

**expenditure**

**Average monthly credit card expenditure.**

**owner**

**Factor. Does the individual own their home?**

**selfemp**

**Factor. Is the individual self-employed?**

**dependents**

**Number of dependents.**

**months**

**Months living at current address.**

**majorcards**

**Number of major credit cards held.**

**active**

**Number of active credit accounts.**

**#import csv files**

Credit\_card<-read.csv(file.choose(),header = T)

View(Credit\_card)

**#display first few records**

head(Credit\_card)

X card reports age income share expenditure owner selfemp dependents months

1 1 yes 0 37.66667 4.5200 0.033269910 124.983300 yes no 3 54

2 2 yes 0 33.25000 2.4200 0.005216942 9.854167 no no 3 34

3 3 yes 0 33.66667 4.5000 0.004155556 15.000000 yes no 4 58

4 4 yes 0 30.50000 2.5400 0.065213780 137.869200 no no 0 25

5 5 yes 0 32.16667 9.7867 0.067050590 546.503300 yes no 2 64

6 6 yes 0 23.25000 2.5000 0.044438400 91.996670 no no 0 54

majorcards active

1 1 12

2 1 13

3 1 5

4 1 7

5 1 5

6 1 1

**#display structure of the data set**

str(Credit\_card)

data.frame': 1319 obs. of 13 variables:

$ X : int 1 2 3 4 5 6 7 8 9 10 ...

$ card : chr "yes" "yes" "yes" "yes" ...

$ reports : int 0 0 0 0 0 0 0 0 0 0 ...

$ age : num 37.7 33.2 33.7 30.5 32.2 ...

$ income : num 4.52 2.42 4.5 2.54 9.79 ...

$ share : num 0.03327 0.00522 0.00416 0.06521 0.06705 ...

$ expenditure: num 124.98 9.85 15 137.87 546.5 ...

$ owner : chr "yes" "no" "yes" "no" ...

$ selfemp : chr "no" "no" "no" "no" ...

$ dependents : int 3 3 4 0 2 0 2 0 0 0 ...

$ months : int 54 34 58 25 64 54 7 77 97 65 ...

$ majorcards : int 1 1 1 1 1 1 1 1 1 1 ...

$ active : int 12 13 5 7 5 1 5 3 6 18 ...

**#display the column names**

colnames(Credit\_card)

[1] "X" "card" "reports" "age" "income" "share"

[7] "expenditure" "owner" "selfemp" "dependents" "months" "majorcards"

[13] "active"

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

sum(is.na(Credit\_card))

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

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

Credit\_card$card<-as.numeric(factor(Credit\_card$card))-1

Credit\_card$owner<-as.numeric(factor(Credit\_card$owner))-1

Credit\_card$selfemp<-as.numeric(factor(Credit\_card$selfemp))-1

str(Credit\_card)

data.frame': 1319 obs. of 13 variables:

$ X : int 1 2 3 4 5 6 7 8 9 10 ...

$ card : num 1 1 1 1 1 1 1 1 1 1 ...

$ reports : int 0 0 0 0 0 0 0 0 0 0 ...

$ age : num 37.7 33.2 33.7 30.5 32.2 ...

$ income : num 4.52 2.42 4.5 2.54 9.79 ...

$ share : num 0.03327 0.00522 0.00416 0.06521 0.06705 ...

$ expenditure: num 124.98 9.85 15 137.87 546.5 ...

$ owner : num 1 0 1 0 1 0 0 1 1 0 ...

$ selfemp : num 0 0 0 0 0 0 0 0 0 0 ...

$ dependents : int 3 3 4 0 2 0 2 0 0 0 ...

$ months : int 54 34 58 25 64 54 7 77 97 65 ...

$ majorcards : int 1 1 1 1 1 1 1 1 1 1 ...

$ active : int 12 13 5 7 5 1 5 3 6 18 ...

**# so we convert all categorical data into numerical form**

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

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

#install.packages("ggplot2")

library(ggplot2)

**#visualization of income**

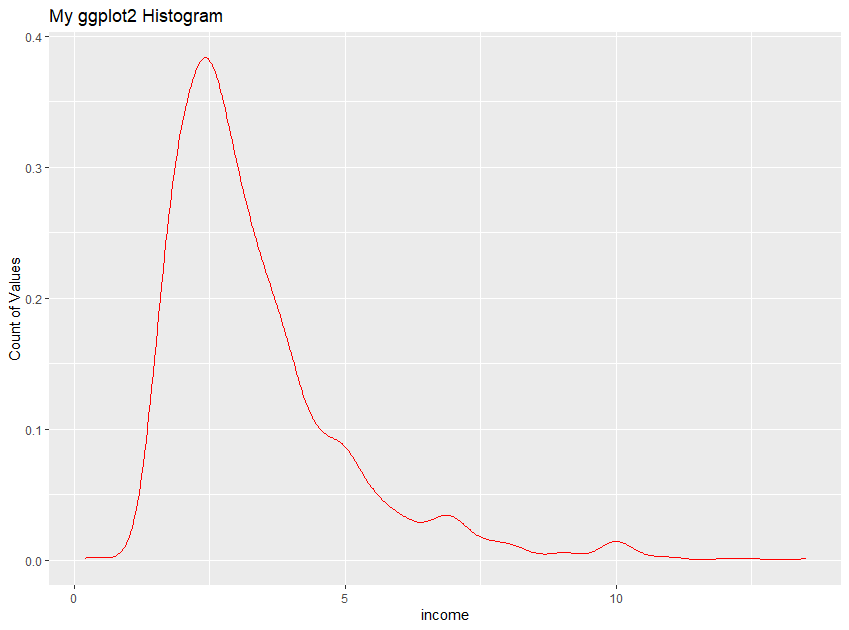
ggplot(Credit\_card, aes(x = income)) +

geom\_density(col = "red") +

labs(title = "My ggplot2 Histogram",

x = "income",

y = "Count of Values")



**#visualization of age**

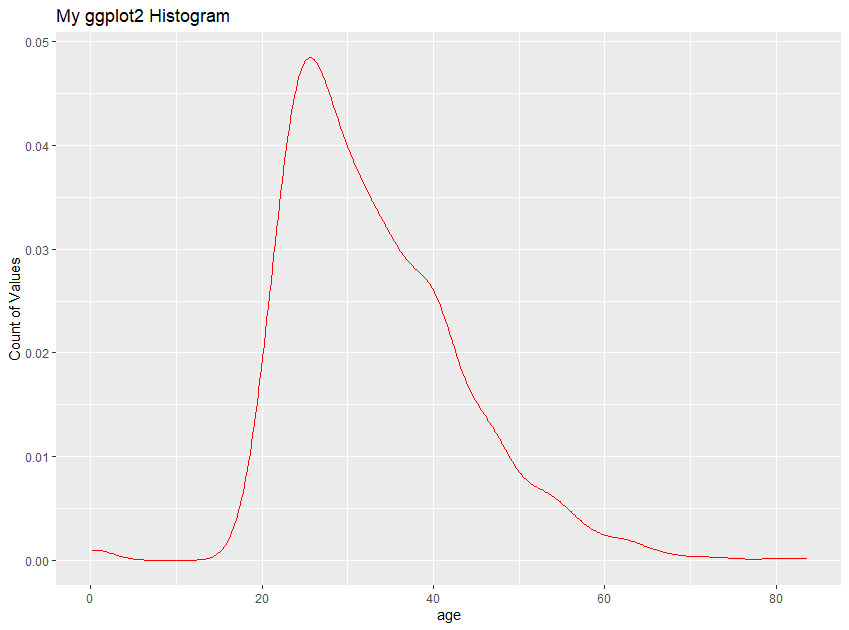
ggplot(Credit\_card, aes(x = age)) +

geom\_density(col = "red") +

labs(title = "My ggplot2 Histogram",

x = "age",

y = "Count of Values")



**#visualize Number of active credit accounts.**

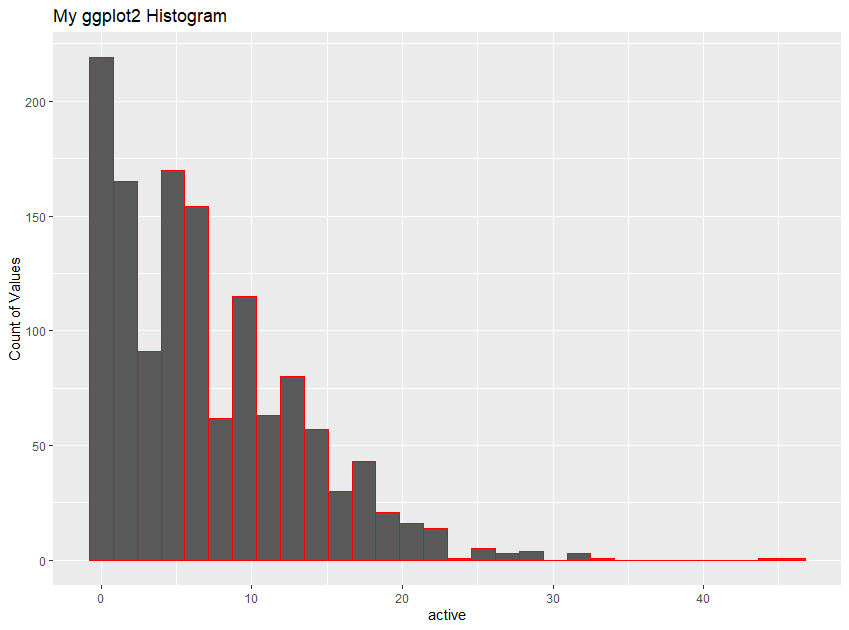
ggplot(Credit\_card, aes(x = active)) +

geom\_histogram(col = "red") +

labs(title = "My ggplot2 Histogram",

x = "active",

y = "Count of Values")



**#visualize Number of dependents**

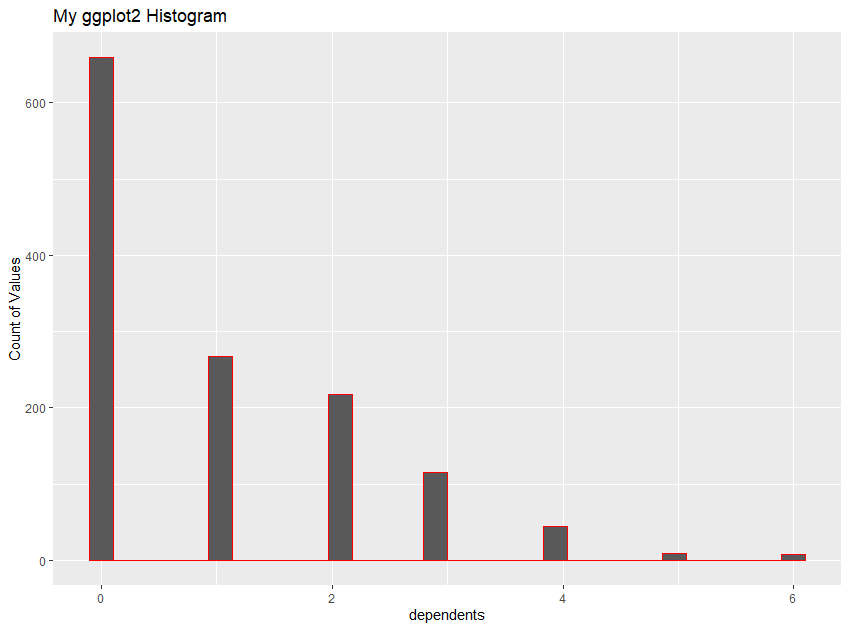
ggplot(Credit\_card, aes(x = dependents)) +

geom\_histogram(col = "red") +

labs(title = "My ggplot2 Histogram",

x = "dependents",

y = "Count of Values")



**#visualize Months living at current address**.

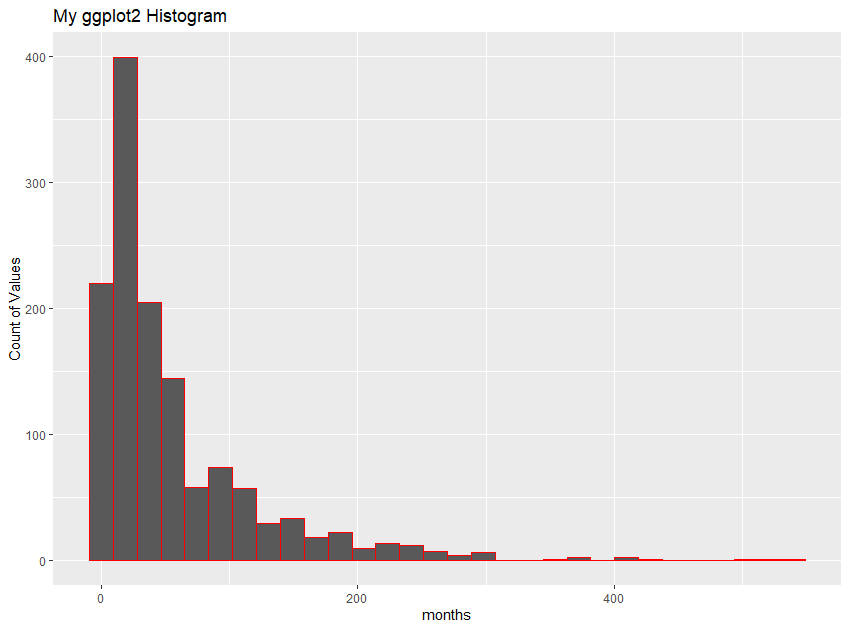
ggplot(Credit\_card, aes(x = months)) +

geom\_histogram(col = "red") +

labs(title = "My ggplot2 Histogram",

x = "months",

y = "Count of Values")



**#visualize the individual self-employed?**

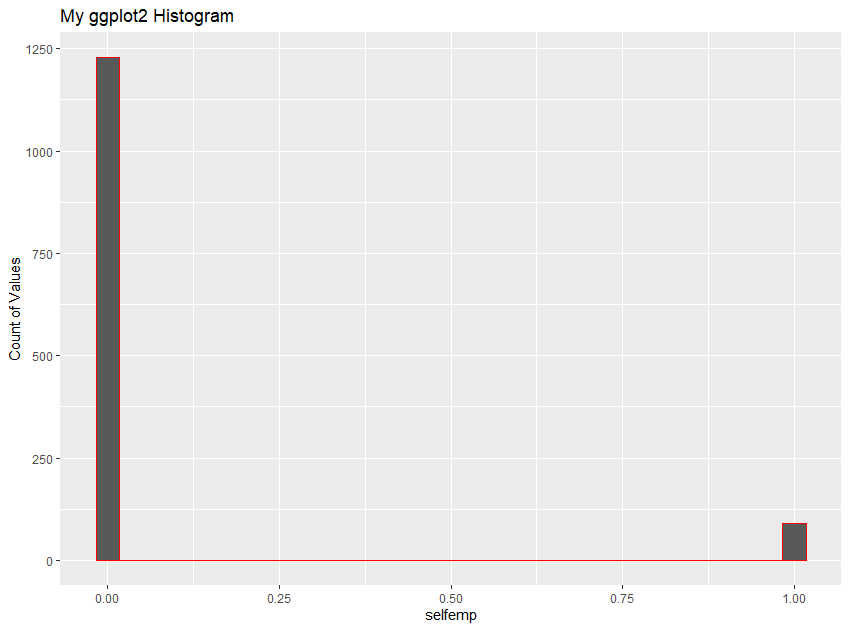
ggplot(Credit\_card, aes(x = selfemp)) +

geom\_histogram(col = "red") +

labs(title = "My ggplot2 Histogram",

x = "selfemp",

y = "Count of Values")



**#here we can see that very less people are self employees**

**#visualize the individual own their home**

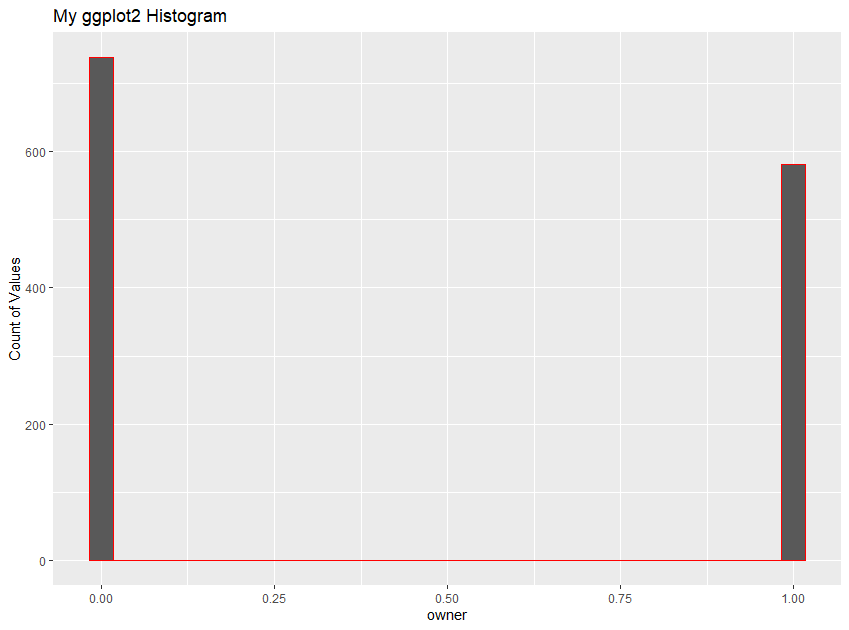
ggplot(Credit\_card, aes(x = owner)) +

geom\_histogram(col = "red") +

labs(title = "My ggplot2 Histogram",

x = "owner",

y = "Count of Values")



**# mejority individulal doesn't have home**

**#visualize Average monthly credit card expenditure**.

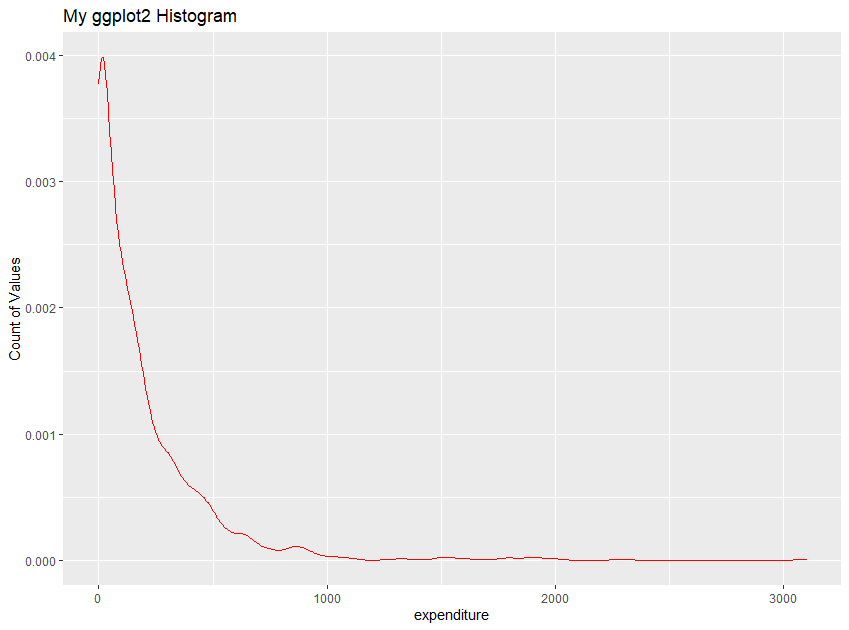
ggplot(Credit\_card, aes(x = expenditure)) +

geom\_density(col = "red") +

labs(title = "My ggplot2 Histogram",

x = "expenditure",

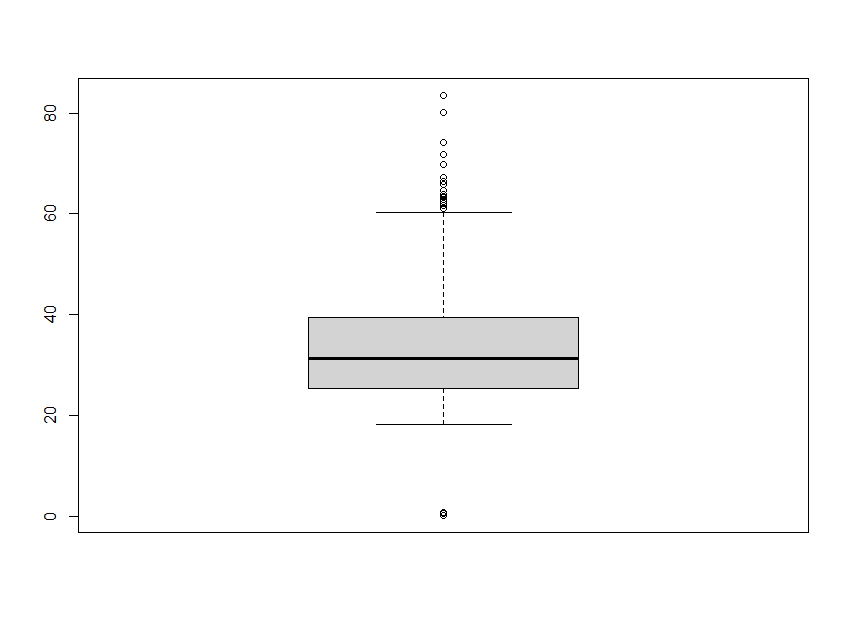
y = "Count of Values")



**#most people credit card expenditure between 500to 1000 us doller**

**#check any ouliers present in our data set using boxplot**

boxplot(Credit\_card$age) #many outliers



boxplot(Credit\_card$income) #many outliers

boxplot(Credit\_card$share) #many outliers

boxplot(Credit\_card$expenditure) #many outliers

boxplot(Credit\_card$months) #many outliers

boxplot(Credit\_card$active) #many outliers

**# so here we detect many outliers, so we have to remove those outliers from our data set**

#here I am removing the outliers from age variables

boxplot(Credit\_card$age,plot = FALSE)$out

[1] 0.5000000 71.8333400 62.9166700 61.0833300 83.5000000 65.9166600 63.5000000 0.1666667

[9] 61.0000000 63.1666700 0.5833333 0.7500000 62.0000000 0.5833333 0.5000000 66.5000000

[17] 74.1666600 64.5833400 62.5000000 80.1666600 61.5833300 61.1666700 61.0833300 63.7500000

[25] 67.1666600 63.4166700 69.7500000 0.7500000

**#removing the outliers from age columns**

outliers <- boxplot(Credit\_card$age, plot=FALSE)$out

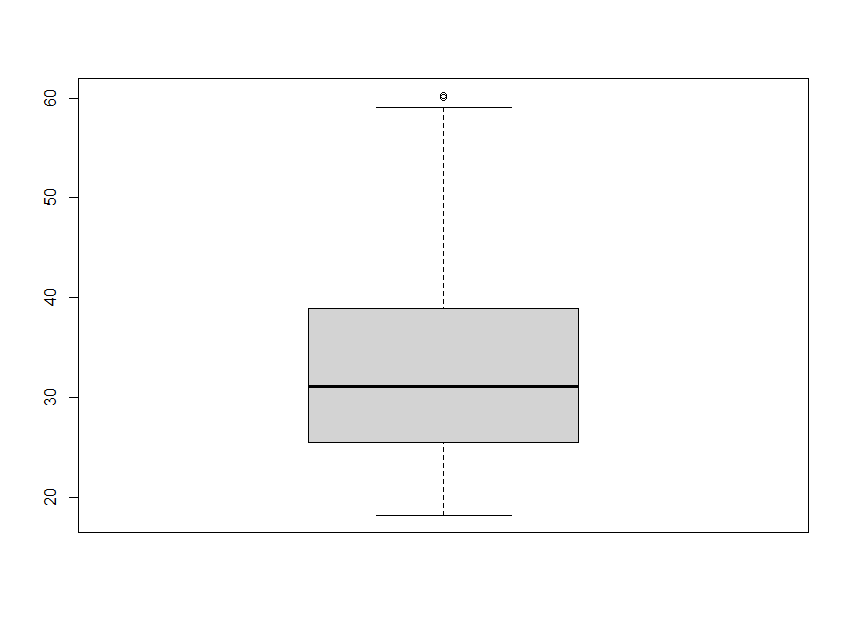
x<- Credit\_card

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

Credit\_card<-x

#box plot after removing the outliers

boxplot(Credit\_card$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**

credit\_data= sample.split(Credit\_card[-1],SplitRatio = 0.3)

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

**#subsetting into Test data**

test\_data =subset(Credit\_card[-1],credit\_data==TRUE)

**#subsetting into Train data**

train\_data=subset(Credit\_card[-1],credit\_data==FALSE)

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

nrow(test\_data)

323

nrow(train\_data)

968

head(train\_data)

|  |
| --- |
| card reports age income share expenditure owner selfemp dependents months  2 1 0 33.25000 2.4200 0.005216942 9.854167 0 0 3 34  3 1 0 33.66667 4.5000 0.004155556 15.000000 1 0 4 58  4 1 0 30.50000 2.5400 0.065213780 137.869200 0 0 0 25  5 1 0 32.16667 9.7867 0.067050590 546.503300 1 0 2 64  7 1 0 27.91667 3.9600 0.012575760 40.833330 0 0 2 7  8 1 0 29.16667 2.3700 0.076433760 150.790000 1 0 0 77  majorcards active  2 1 13  3 1 5  4 1 7  5 1 5  7 1 5  8 1 3 |
|  |
| |  | | --- | |  | |

head(test\_data)

card reports age income share expenditure owner selfemp dependents months

1 1 0 37.66667 4.52 0.0332699100 124.98330 1 0 3 54

6 1 0 23.25000 2.50 0.0444384000 91.99667 0 0 0 54

11 1 0 30.50000 3.95 0.0780245600 256.66420 1 0 1 24

13 0 0 30.00000 1.73 0.0006936416 0.00000 1 0 1 42

18 0 7 29.50000 3.00 0.0004000000 0.00000 1 0 2 60

23 1 0 34.25000 2.00 0.1311120000 218.52000 1 0 0 12

majorcards active

1 1 12

6 1 1

11 1 20

13 0 12

18 1 8

23 1 0

View((train\_data))

View((test\_data))

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

**#BUILDS A MODEL**

**#Classify whether application accepted or not using Logistic regression**

**# here "card" is our target variable**

credit\_model <- glm(card~.,data=train\_data,family = "binomial")

summary(credit\_model)

Call:

glm(formula = card ~ ., family = "binomial", data = train\_data)

Deviance Residuals:

Min 1Q Median 3Q Max

-8.49 0.00 0.00 0.00 8.49

Coefficients:

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

(Intercept) 3.554e+14 1.063e+07 33432472 <2e-16 \*\*\*

reports -6.414e+14 1.742e+06 -368227388 <2e-16 \*\*\*

age 1.095e+13 2.951e+05 37119753 <2e-16 \*\*\*

income -3.179e+13 1.791e+06 -17752039 <2e-16 \*\*\*

share 1.731e+16 5.184e+07 333974970 <2e-16 \*\*\*

expenditure -7.092e+11 1.803e+04 -39334422 <2e-16 \*\*\*

owner -7.280e+13 5.134e+06 -14179281 <2e-16 \*\*\*

selfemp 3.906e+14 8.887e+06 43949660 <2e-16 \*\*\*

dependents -1.442e+13 1.916e+06 -7524161 <2e-16 \*\*\*

months -1.625e+12 3.963e+04 -41015304 <2e-16 \*\*\*

majorcards 6.250e+13 5.730e+06 10907245 <2e-16 \*\*\*

active 9.498e+12 3.751e+05 25321686 <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: 1017.7 on 967 degrees of freedom

Residual deviance: 8290.0 on 956 degrees of freedom

AIC: 8314

Number of Fisher Scoring iterations: 25

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

credit\_prob <- predict(credit\_model,type=c("response"),train\_data)

View(credit\_prob)

|  |
| --- |
| head(credit\_prob)  2 3 4 5 7 8  1 1 1 1 1 1 |
|  |
| |  | | --- | |  | |

**###create confusion matrix table**

conf\_matrix<-table(credit\_prob>0.5,train\_data$card)

conf\_matrix

|  |
| --- |
| 0 1  FALSE 101 4  TRUE 111 752 |
|  |
| |  | | --- | | > | |

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

**#check the Model Accuracy**

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

**Accuracy**

# 98.55% of accuracy

**#check the error rate**

1-Accuracy

#0.01% error rate. very less 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.**

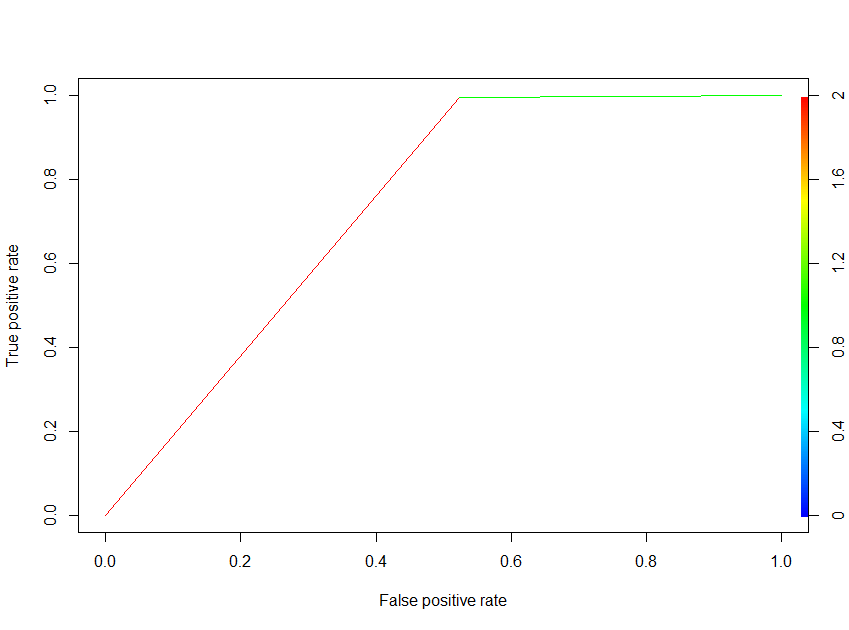
#install.packages("ROCR")

library(ROCR)

roc\_prediction<-prediction(credit\_prob,train\_data$card)

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.**

**# This is a perfect curve. When two curves don’t overlap at all means model**

**#has an ideal measure of separability.**

**#It is perfectly able to distinguish between positive class and negative class**.

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

**#Predict on Test data set**

test\_prediction <- predict(credit\_model, newdata = test\_data, type = "response")

test\_pred\_num <- ifelse(test\_prediction > 0.5, 1, 0)

pred\_test <- factor(test\_pred\_num, levels=c(0, 1))

test\_actual\_data<-test\_data$card

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

head(test\_data)

|  |
| --- |
| card reports age income share expenditure owner selfemp dependents months  1 1 0 37.66667 4.52 0.0332699100 124.98330 1 0 3 54  6 1 0 23.25000 2.50 0.0444384000 91.99667 0 0 0 54  11 1 0 30.50000 3.95 0.0780245600 256.66420 1 0 1 24  13 0 0 30.00000 1.73 0.0006936416 0.00000 1 0 1 42  18 0 7 29.50000 3.00 0.0004000000 0.00000 1 0 2 60  23 1 0 34.25000 2.00 0.1311120000 218.52000 1 0 0 12  majorcards active  1 1 12  6 1 1  11 1 20  13 0 12  18 1 8  23 1 0 |
|  |
| |  | | --- | | > | |

test\_data$predict\_data<- pred\_test

head(test\_data)

card reports age income share expenditure owner selfemp dependents months

1 1 0 37.66667 4.52 0.0332699100 124.98330 1 0 3 54

6 1 0 23.25000 2.50 0.0444384000 91.99667 0 0 0 54

11 1 0 30.50000 3.95 0.0780245600 256.66420 1 0 1 24

13 0 0 30.00000 1.73 0.0006936416 0.00000 1 0 1 42

18 0 7 29.50000 3.00 0.0004000000 0.00000 1 0 2 60

23 1 0 34.25000 2.00 0.1311120000 218.52000 1 0 0 12

majorcards active predict\_data

1 1 12 1

6 1 1 1

11 1 20 1

13 0 12 1

18 1 8 0

23 1 0 1

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

mean(pred\_test== test\_actual\_data) **#98.44 %accuracy**

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

View(new\_test\_df)

#displaying first 20 data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | head(new\_test\_df,20)  card predict\_data  1 1 1  6 1 1  11 1 1  13 0 1  18 0 0  23 1 1  25 1 1  30 1 1  35 1 1  37 0 1  42 1 1  47 1 1  49 1 1  54 1 1  59 0 1  61 1 0  66 0 1  71 1 1  73 1 1  78 1 1 | |  | | |  | | --- | | > | | |
|  |
|  |

**#Conclsion:**

**# here we have created logistic model using train data set with 98.55 accuracy**

**#using this model correctly, predict whether application accepted or not using test data set**