FORTUNE HOUSING FINANCE COMPANY

A Company wants to automate the loan eligibility process (real time) based on customer detail provided while filling online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others. To automate this process, they have given a problem to identify the customers segments, those are eligible for loan amount so that they can specifically target these customers. They have provided a partial data set. This loan dataset consists of 614 instances and 13 attributes.

Primarily, I am trying to analyze what factors are contributing to the Loan approval at Fortune finance company like will the customers with good credit history or customers who has property in urban, semi urban areas etc. Since I am trying to analyze the Loan status and its relation to other related variables associated with it I first choose to use “Multiple Linear Regression”, as this model helps to establish a linear relationship between a response variable(Loan status) and Predictor variables(Like Credit History, Gender, Education etc.)

In the second part I am trying to predict the Loan Status of the applicants which helps the business to know more about their targeted customers. Here as I want to predict the Loan status of applicants with the variables I have, using LOGISTIC REGRESSION algorithm will be more appropriate.

 # Required packages and libraries used for the project in R

install.packages("dplyr")

library(dplyr)

install.packages("plyr")

library(plyr)

install.packages("ggplot2")

library(ggplot2)

install.packages(c("corrplot"))

library(corrplot)

install.packages(c("ggm", "gmodels", "vcd", "Hmisc", "pastecs", "psych", "doBy"))

library(ggm)

library(Hmisc)

library(pastecs)

library(psych)

library(doBy)

library(vcd)

```
library(gmodels)
install.packages("car")
library(car)
install.packages('MASS')
library(MASS)
# This command is used to get the location of current working directory
getwd()
# This command is used to point to the folder containing the required file
setwd("U:/R/R project/Final Project")
#Read the file Loan.csv
#This command imports the required data set and saves it to the Loan data frame.
Loan <- read.csv("Loan.csv",header=TRUE)
Loan
```

<u>COLLECTING THE DATASET</u>

(Attribute description):

Variable	Description
Loan_ID	Unique Loan ID
Gender	Male/ Female
Married	Applicant married (Yes/No)
Dependents	Number of dependents (0-3)
Education	Applicant Education (Graduate/ Not Graduate)
Self_Employed	Self-employed (Yes/No)
ApplicantIncome	Applicant income (per month in Rupees)
CoapplicantIncome	Coapplicant income (per month in Rupees)
LoanAmount	Loan amount in thousands (of Rupees)
Loan_Amount_Term	Term of loan in months
Credit_History	Credit history meets guidelines (1,0)
Property_Area	Urban/ Semi Urban/ Rural
Loan_Status	Loan approved (Y/N)

Sample dataset:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
1	LP001002	Male	No	0	Graduate	No	5849	0	NA	360	1	Urban	Y
2	LP001003	Male	Yes	1	Graduate	No	4583	1508	128	360	1	Rural	N
3	LP001005	Male	Yes	0	Graduate	Yes	3000	0	66	360	1	Urban	Y
4	LP001006	Male	Yes	0	Not Graduate	No	2583	2358	120	360	1	Urban	Y
5	LP001008	Male	No	0	Graduate	No	6000	0	141	360	1	Urban	Y
6	LP001011	Male	Yes	2	Graduate	Yes	5417	4196	267	360	1	Urban	Y
7	LP001013	Male	Yes	0	Not Graduate	No	2333	1516	95	360	1	Urban	Y
8	LP001014	Male	Yes	3	Graduate	No	3036	2504	158	360	0	Semiurban	N
9	LP001018	Male	Yes	2	Graduate	No	4006	1526	168	360	1	Urban	Y
10	LP001020	Male	Yes	1	Graduate	No	12841	10968	349	360	1	Semiurban	N
11	LP001024	Male	Yes	2	Graduate	No	3200	700	70	360	1	Urban	Y
12	LP001027	Male	Yes	2	Graduate	No	2500	1840	109	360	1	Urban	Y
13	LP001028	Male	Yes	2	Graduate	No	3073	8106	200	360	1	Urban	Y
14	LP001029	Male	No	0	Graduate	No	1853	2840	114	360	1	Rural	N
15	LP001030	Male	Yes	2	Graduate	No	1299	1086	17	120	1	Urban	Y
16	LP001032	Male	No	0	Graduate	No	4950	0	125	360	1	Urban	Y
17	LP001034	Male	No	1	Not Graduate	No	3596	0	100	240	0	Urban	Y
18	LP001036	Female	No	0	Graduate	No	3510	0	76	360	0	Urban	N
19	LP001038	Male	Yes	0	Not Graduate	No	4887	0	133	360	1	Rural	N
20	LP001041	Male	Yes	0	Graduate	No	2600	3500	115	NA	1	Urban	Y
21	LP001043	Male	Yes	0	Not Graduate	No	7660	0	104	360	0	Urban	N
22	LP001046	Male	Yes	1	Graduate	No	5955	5625	315	360	1	Urban	Y
23	LP001047	Male	Yes	0	Not Graduate	No	2600	1911	116	360	0	Semiurban	N
24	LP001050	Female	Yes	2	Not Graduate	No	3365	1917	112	360	0	Rural	N
25	LP001052	Male	Yes	1	Graduate	No	3717	2925	151	360	0	Semiurban	N

PREPARATION AND CONVERSION OF THE DATA

 # Structure of Loan data frame to see if the data is structured or not

str(Loan)

Console Terminal

U:/R/R project/Final Project/

```
> str(Loan)
'data.frame':	614 obs. of  13 variables:
 $ Loan_ID          : Factor w/ 614 levels "LP001002","LP001003",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ Gender           : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
 $ Married          : Factor w/ 2 levels "No","Yes": 1 2 2 2 1 2 2 2 2 2 ...
 $ Dependents       : int  0 1 0 0 0 2 0 3 2 1 ...
 $ Education        : Factor w/ 2 levels "Graduate","Not Graduate": 1 1 1 2 1 1 2 1 1 1 ...
 $ Self_Employed    : Factor w/ 2 levels "No","Yes": 1 1 2 1 1 2 1 1 1 1 ...
 $ ApplicantIncome  : int  5849 4583 3000 2583 6000 5417 2333 3036 4006 12841 ...
 $ CoapplicantIncome: num  0 1508 0 2358 0 ...
 $ LoanAmount       : int  NA 128 66 120 141 267 95 158 168 349 ...
 $ Loan_Amount_Term : int  360 360 360 360 360 360 360 360 360 360 ...
 $ Credit_History   : int  1 1 1 1 1 1 1 0 1 1 ...
 $ Property_Area    : Factor w/ 3 levels "Rural","Semiurban",..: 3 1 3 3 3 3 3 2 3 2 ...
 $ Loan_Status      : Factor w/ 2 levels "N","Y": 2 1 2 2 2 2 2 1 2 1 ...
> 
```

#Checking for the structure and other possible incompleteness

summary(Loan)

```
Console   Terminal ×
U:/R/R project/Final Project/
> summary(Loan)
     Loan_ID       Gender      Married      Dependents            Education    Self_Employed ApplicantIncome
 LP001002:  1   Female:125   No :216    Min.   :0.0000    Graduate     :480   No :532       Min.   :  150
 LP001003:  1   Male  :489   Yes:398    1st Qu.:0.0000    Not Graduate:134   Yes: 82       1st Qu.: 2878
 LP001005:  1                           Median :0.0000                                     Median : 3812
 LP001006:  1                           Mean   :0.7443                                     Mean   : 5403
 LP001008:  1                           3rd Qu.:1.0000                                     3rd Qu.: 5795
 LP001011:  1                           Max.   :3.0000                                     Max.   :81000
 (Other) :608
 CoapplicantIncome   LoanAmount     Loan_Amount_Term Credit_History    Property_Area Loan_Status
 Min.   :    0     Min.   :  9.0   Min.   : 12     Min.   :0.0000   Rural    :179   N:192
 1st Qu.:    0     1st Qu.:100.0   1st Qu.:360     1st Qu.:1.0000   Semiurban:233   Y:422
 Median : 1188     Median :128.0   Median :360     Median :1.0000   Urban    :202
 Mean   : 1621     Mean   :146.4   Mean   :342     Mean   :0.7736
 3rd Qu.: 2297     3rd Qu.:168.0   3rd Qu.:360     3rd Qu.:1.0000
 Max.   :41667     Max.   :700.0   Max.   :480     Max.   :1.0000
                   NA's   :22      NA's   :14
> |
```

sum(is.na(Loan))

```
Console   Terminal ×
U:/R/R project/Final Project/
> sum(is.na(Loan))
[1] 36
```

#The data set now has 36 missing values.

#Replacing the NA valus with mean values of the Loan Amount Variable

Loan$LoanAmount <- ifelse(is.na(Loan$LoanAmount),

ave(Loan$LoanAmount,FUN = function(x)mean(x,na.rm=TRUE)),

Loan$LoanAmount)

#Replacing the NA valus with mean values of the Loan Amount Term

Loan$Loan_Amount_Term <-ifelse(is.na(Loan$Loan_Amount_Term),

ave(Loan$Loan_Amount_Term,FUN = function(x)mean(x,na.rm=TRUE)),

Loan$Loan_Amount_Term)

#Credit History is described whether or not customer meets guidelines.

#Loan Status 1 for approved loan, 0 for rejected.

Loan$Credit_History = factor(Loan$Credit_History, levels = c(0,1), labels = c("Unmet", "Met"))

Loan$Loan_Status = as.numeric(Loan$Loan_Status) – 1

LoanData<-Loan

#save the file in our current working directory

write.table(LoanData,file="LoanData.csv",row.names=F,sep=",")

summary(LoanData)

```
Console   Terminal ×
U:/R/R project/Final Project/
> summary(LoanData)
     Loan_ID       Gender      Married      Dependents          Education     Self_Employed ApplicantIncome CoapplicantIncome
 LP001002:  1   Female:125   No :216   Min.   :0.0000   Graduate     :480   No :532       Min.   :  150   Min.   :    0
 LP001003:  1   Male  :489   Yes:398   1st Qu.:0.0000   Not Graduate:134   Yes: 82       1st Qu.: 2878   1st Qu.:    0
 LP001005:  1                          Median :0.0000                                    Median : 3812   Median : 1188
 LP001006:  1                          Mean   :0.7443                                    Mean   : 5403   Mean   : 1621
 LP001008:  1                          3rd Qu.:1.0000                                    3rd Qu.: 5795   3rd Qu.: 2297
 LP001011:  1                          Max.   :3.0000                                    Max.   :81000   Max.   :41667
 (Other) :608
   LoanAmount     Loan_Amount_Term Credit_History   Property_Area   Loan_Status
 Min.   :  9.0   Min.   : 12      Unmet:139      Rural    :179   Min.   :0.0000
 1st Qu.:100.2   1st Qu.:360      Met  :475      Semiurban:233   1st Qu.:0.0000
 Median :129.0   Median :360                     Urban    :202   Median :1.0000
 Mean   :146.4   Mean   :342                                     Mean   :0.6873
 3rd Qu.:164.8   3rd Qu.:360                                     3rd Qu.:1.0000
 Max.   :700.0   Max.   :480                                     Max.   :1.0000

> |
```

str(LoanData)

```
Console   Terminal ×
U:/R/R project/Final Project/

> str(LoanData)
'data.frame':	614 obs. of  13 variables:
 $ Loan_ID          : Factor w/ 614 levels "LP001002","LP001003",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ Gender           : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
 $ Married          : Factor w/ 2 levels "No","Yes": 1 2 2 2 1 2 2 2 2 2 ...
 $ Dependents       : int  0 1 0 0 0 2 0 3 2 1 ...
 $ Education        : Factor w/ 2 levels "Graduate","Not Graduate": 1 1 1 2 1 1 2 1 1 1 ...
 $ Self_Employed    : Factor w/ 2 levels "No","Yes": 1 1 2 1 1 2 1 1 1 1 ...
 $ ApplicantIncome  : int  5849 4583 3000 2583 6000 5417 2333 3036 4006 12841 ...
 $ CoapplicantIncome: num  0 1508 0 2358 0 ...
 $ LoanAmount       : num  146 128 66 120 141 ...
 $ Loan_Amount_Term : num  360 360 360 360 360 360 360 360 360 360 ...
 $ Credit_History   : Factor w/ 2 levels "Unmet","Met": 2 2 2 2 2 2 2 1 2 2 ...
 $ Property_Area    : Factor w/ 3 levels "Rural","Semiurban",..: 3 1 3 3 3 3 3 2 3 2 ...
 $ Loan_Status      : num  1 0 1 1 1 1 1 0 1 0 ...
> |
```

sum(is.na(LoanData))

```
Console   Terminal ×
U:/R/R project/Final Project/
> sum(is.na(LoanData))
[1] 0
> |
```

#The data set now has 0 missing values.

CONTINGENCY TABLES FOR COMPARISON

#One-way contingency tables for the categorical variables.

 # We can create simple frequency counts using the table() function in base R

#GENDER

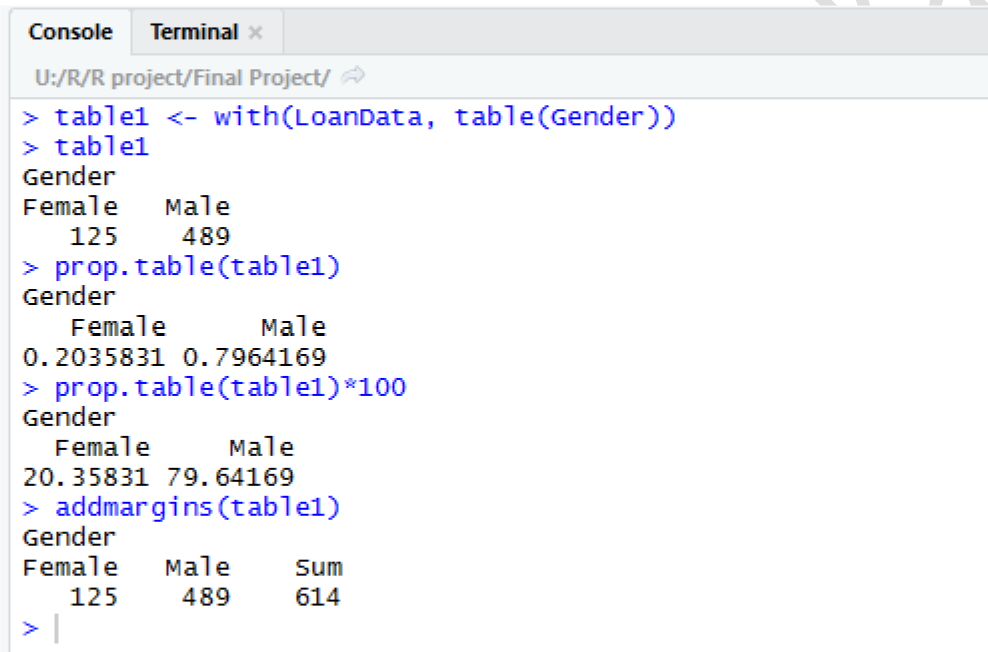
table1 <- with(LoanData, table(Gender))

table1# frequencies

prop.table(table1)# proportions

*prop.table(table1)*100 # percentages*

addmargins(table1)



#MARRIED

table2 <- with(LoanData, table(Married))

table2# frequencies

prop.table(table2)# proportions

*prop.table(table2)*100 # percentages*

addmargins(table2)

```
Console   Terminal ×
U:/R/R project/Final Project/
> table2 <- with(LoanData, table(Married))
> table2# frequencies
Married
 No Yes
216 398
> prop.table(table2)# proportions
Married
       No       Yes
0.3517915 0.6482085
> prop.table(table2)*100 # percentages
Married
      No      Yes
35.17915 64.82085
> addmargins(table2)
Married
 No Yes Sum
216 398 614
> |
```

#EDUCATION

table3 <- with(LoanData, table(Education))

table3# frequencies

prop.table(table3)# proportions

*prop.table(table3)*100 # percentages*

addmargins(table3)

```
Console   Terminal ×
U:/R/R project/Final Project/
> table3 <- with(LoanData, table(Education))
> table3# frequencies
Education
    Graduate Not Graduate
         480          134
> prop.table(table3)# proportions
Education
    Graduate Not Graduate
    0.781759     0.218241
> prop.table(table3)*100 # percentages
Education
    Graduate Not Graduate
     78.1759      21.8241
> addmargins(table3)
Education
    Graduate Not Graduate          Sum
         480          134          614
> |
```

#SELF-EMPLOYED

table4 <- with(LoanData, table(Self_Employed))

table4# frequencies

prop.table(table4)# proportions

*prop.table(table4)*100 # percentages*

addmargins(table4)

```
Console   Terminal ×
U:/R/R project/Final Project/
> table4 <- with(LoanData, table(Self_Employed))
> table4# frequencies
Self_Employed
 No Yes
532  82
> prop.table(table4)# proportions
Self_Employed
       No       Yes
0.8664495 0.1335505
> prop.table(table4)*100 # percentages
Self_Employed
      No      Yes
86.64495 13.35505
> addmargins(table4)
Self_Employed
 No Yes Sum
532  82 614
>
```

#CREDIT_HISTORY

table5 <- with(LoanData, table(Credit_History))

table5# frequencies

prop.table(table5)# proportions

*prop.table(table5)*100 # percentages*

addmargins(table5)

```
Console   Terminal ×
U:/R/R project/Final Project/
> table5 <- with(LoanData, table(Credit_History))
> table5# frequencies
Credit_History
Unmet   Met
  139   475
> prop.table(table5)# proportions
Credit_History
    Unmet       Met
0.2263844 0.7736156
> prop.table(table5)*100 # percentages
Credit_History
   Unmet      Met
22.63844 77.36156
> addmargins(table5)
Credit_History
Unmet   Met   Sum
  139   475   614
>
```

#PROPERTY-AREA

table6 <- with(LoanData, table(Property_Area))

table6# frequencies

prop.table(table6)# proportions

*prop.table(table6)*100 # percentages*

addmargins(table6)

```
Console   Terminal ×
U:/R/R project/Final Project/
> table6 <- with(LoanData, table(Property_Area))
> table6# frequencies
Property_Area
    Rural Semiurban     Urban
      179       233       202
> prop.table(table6)# proportions
Property_Area
    Rural Semiurban     Urban
0.2915309 0.3794788 0.3289902
> prop.table(table6)*100 # percentages
Property_Area
    Rural Semiurban     Urban
 29.15309  37.94788  32.89902
> addmargins(table6)
Property_Area
    Rural Semiurban     Urban       Sum
      179       233       202       614
> |
```

#Two-way contingency tables for the categorical variables

 # Alternatively, the xtabs() function allows you to create a contingency

 # table using formula style input

#LOAN-STATUS & CREDIT-HISTORY

table7 <- xtabs(~ Loan_Status+Credit_History, data=LoanData)

table7

addmargins(table7)

```
Console   Terminal ×
U:/R/R project/Final Project/
> #LOAN-STATUS & CREDIT-HISTORY
> table7 <- xtabs(~ Loan_Status+Credit_History, data=LoanData)
> table7
           Credit_History
Loan_Status Unmet Met
          0    95  97
          1    44 378
> addmargins(table7)
           Credit_History
Loan_Status Unmet Met Sum
        0      95  97 192
        1      44 378 422
        Sum   139 475 614
> |
```

#LOAN-STATUS & PROPERTY-AREA

table8 <- xtabs(~ Loan_Status+Property_Area, data=LoanData)

table8

addmargins(table8)

```
Console   Terminal ×
U:/R/R project/Final Project/
> table8 <- xtabs(~ Loan_Status+Property_Area, data=LoanData)
> table8
           Property_Area
Loan_Status Rural Semiurban Urban
          0    69        54    69
          1   110       179   133
> addmargins(table8)
           Property_Area
Loan_Status Rural Semiurban Urban Sum
        0      69        54    69 192
        1     110       179   133 422
        Sum   179       233   202 614
> |
```

#LOAN-STATUS & SELF-EMPLOYED

table9 <- xtabs(~ Loan_Status+Self_Employed, data=LoanData)

table9

addmargins(table9)

```
Console   Terminal ×
U:/R/R project/Final Project/
> table9 <- xtabs(~ Loan_Status+Self_Employed, data=LoanData)
> table9
           Self_Employed
Loan_Status  No Yes
          0 166  26
          1 366  56
> addmargins(table9)
           Self_Employed
Loan_Status  No Yes Sum
        0   166  26 192
        1   366  56 422
        Sum 532  82 614
> 
```

#LOAN-STATUS & EDUCATION

table10 <- xtabs(~ Loan_Status+Education, data=LoanData)

table10

addmargins(table10)

```
Console   Terminal ×
U:/R/R project/Final Project/
> table10 <- xtabs(~ Loan_Status+Education, data=LoanData)
> table10
           Education
Loan_Status Graduate Not Graduate
          0      140           52
          1      340           82
> addmargins(table10)
           Education
Loan_Status Graduate Not Graduate Sum
        0        140           52 192
        1        340           82 422
        Sum      480          134 614
> 
```

#LOAN-STATUS & MARRIED

table11 <- xtabs(~ Loan_Status+Married, data=LoanData)

table11

addmargins(table11)

```
Console   Terminal ×
U:/R/R project/Final Project/
> table11 <- xtabs(~ Loan_Status+Married, data=LoanData)
> table11
           Married
Loan_Status  No Yes
          0  79 113
          1 137 285
> addmargins(table11)
           Married
Loan_Status  No Yes Sum
        0    79 113 192
        1   137 285 422
        Sum 216 398 614
>
```

#LOAN-STATUS & GENDER

table12 <- xtabs(~ Loan_Status+Gender, data=LoanData)

table12

addmargins(table12)

```
Console   Terminal ×
U:/R/R project/Final Project/
> table12 <- xtabs(~ Loan_Status+Gender, data=LoanData)
> table12
           Gender
Loan_Status Female Male
          0     42  150
          1     83  339
> addmargins(table12)
           Gender
Loan_Status Female Male Sum
        0       42  150 192
        1       83  339 422
        Sum    125  489 614
>
```

<u>CHI-SQUARE TEST OF INDEPENDENCE</u>

#It is used to determine whether there is a significant association between the two variables.

#The chi-square goodness of fit test is appropriate when the following conditions are met:

a) *The sampling method is simple random sampling.*
b) *The variable under study is categorical.*

#Degrees of freedom: The degrees of freedom (DF) is equal to the number of levels (k) of the categorical variable minus 1.

 # DF = k - 1

*#**Expected frequency count**s: The expected frequency counts at each level of the categorical variable*

#are equal to the sample size times the hypothesized proportion from the null hypothesis

 # Ei = npi

#where Ei is the expected frequency count for the ith level of the categorical variable,

#n is the total sample size, and pi is the hypothesized proportion of observations in level i.

#Test statistic: The test statistic is a chi-square random variable defined by the following equation.

 # chi-square=[(Oi - Ei)2 / Ei]

#where Oi is the observed frequency count for the ith level of the categorical variable,

#and Ei is the expected frequency count for the ith level of the categorical variable.

#P-value: The P-value is the probability of observing a sample statistic as extreme as the test statistic

#H0: Variables X and Loan Status are independent

#Ha: Variables X and Loan Status are not independent

#If the P-value is less than the significance level (0.05), we cannot accept the null hypothesis.

#so, if p>0.05 then that Loan status is independent of that variable and need not consider

#that variable for further analysis.

#FOR GENDER VARIABLE

chisq.test(LoanData$Gender,LoanData$Loan_Status)

```
Console   Terminal ×
U:/R/R project/Final Project/
> chisq.test(LoanData$Gender,LoanData$Loan_Status)

        Pearson's Chi-squared test with Yates' continuity correction

data:  LoanData$Gender and LoanData$Loan_Status
X-squared = 0.27192, df = 1, p-value = 0.602

>
```

#We can say that Loan approval doesn't depend on gender

#FOR MARRIED VARIABLE

chisq.test(LoanData$Married,LoanData$Loan_Status)

```
Console   Terminal ×
U:/R/R project/Final Project/
> chisq.test(LoanData$Married,LoanData$Loan_Status)

        Pearson's Chi-squared test with Yates' continuity correction

data:  LoanData$Married and LoanData$Loan_Status
X-squared = 3.989, df = 1, p-value = 0.0458

> 
```

#It's apparent that Loan approval depends on Marital status

#FOR NO.OF DEPENDENTS VARIABLE

chisq.test(LoanData$Dependents,LoanData$Loan_Status)

```
Console   Terminal ×
U:/R/R project/Final Project/
> chisq.test(LoanData$Dependents,LoanData$Loan_Status)

        Pearson's Chi-squared test

data:  LoanData$Dependents and LoanData$Loan_Status
X-squared = 3.1514, df = 3, p-value = 0.3689

> 
```

#We can say that Loan approval doesn't depend on Number of Dependents

#FOR EDUCATION VARIABLE

chisq.test(LoanData$Education,LoanData$Loan_Status)

```
Console   Terminal ×
U:/R/R project/Final Project/
> chisq.test(LoanData$Education,LoanData$Loan_Status)

        Pearson's Chi-squared test with Yates' continuity correction

data:  LoanData$Education and LoanData$Loan_Status
X-squared = 4.0915, df = 1, p-value = 0.0431

> 
```

#It's apparent that Loan approval depends on Education

#FOR SELF-EMPLOYED VARIABLE

chisq.test(LoanData$Self_Employed,LoanData$Loan_Status)

```
Console   Terminal ×
U:/R/R project/Final Project/
> chisq.test(LoanData$Self_Employed,LoanData$Loan_Status)

        Pearson's Chi-squared test with Yates' continuity correction

data:  LoanData$Self_Employed and LoanData$Loan_Status
X-squared = 1.0223e-29, df = 1, p-value = 1

> |
```

#Loan approval doesn't depend on if applicant is self employed

#FOR CREDIT-HISTORY VARIABLE

chisq.test(LoanData$Credit_History,LoanData$Loan_Status)

```
Console   Terminal ×
U:/R/R project/Final Project/
> chisq.test(LoanData$Credit_History,LoanData$Loan_Status)

        Pearson's Chi-squared test with Yates' continuity correction

data:  LoanData$Credit_History and LoanData$Loan_Status
X-squared = 112.7, df = 1, p-value < 2.2e-16

> |
```

#It's apparent that Loan approval depends on Credit History

#FOR PROPERTY_AREA VARIABLE

chisq.test(LoanData$Property_Area,LoanData$Loan_Status)

```
Console   Terminal ×
U:/R/R project/Final Project/
> chisq.test(LoanData$Property_Area,LoanData$Loan_Status)

        Pearson's Chi-squared test

data:  LoanData$Property_Area and LoanData$Loan_Status
X-squared = 12.298, df = 2, p-value = 0.002136

> |
```

#It's apparent that Loan approval depends on Property area.

#Referring to p.values located in the table above, we can conclude that following significance level of 5% variables:

 # Gender

#Dependents

#Self_Employed

#are independent of loan_status and therefore should give small predictive power in future model.

#FOR LOAN AMOUNT VARIABLE

chisq.test(LoanData$LoanAmount,LoanData$Loan_Status)

```
Console   Terminal ×
U:/R/R project/Final Project/
> chisq.test(LoanData$LoanAmount,LoanData$Loan_Status)

        Pearson's Chi-squared test

data:  LoanData$LoanAmount and LoanData$Loan_Status
X-squared = 205.53, df = 203, p-value = 0.4373

Warning message:
In chisq.test(LoanData$LoanAmount, LoanData$Loan_Status) :
  Chi-squared approximation may be incorrect
> |
```

#Since the chi-square can't perform well on Loan Amount Variable

#I started performing various Statistical analysis on continuous variables.

<u>#STATISTICAL ANALYSIS ON CONTINUOUS VARIABLES</u>

```
mystats <- function(x, na.omit=FALSE){
 if (na.omit)
  x <- x[!is.na(x)]
 m <- mean(x)
 mi<-min(x)
 ma<-max(x)
 me<-median(x)
```

IQR<-IQR(x,na.rm=FALSE,type=7)

n <- length(x)

s <- sd(x)

skew <- sum((x-m)^3/s^3)/n

kurt <- sum((x-m)^4/s^4)/n - 3

return(c(length=n, min=mi, max=ma, median=me, mean=m, IQR=IQR, stdev=s, skew=skew, kurtosis=kurt))

}

#Applicant Income

myvars <- c("ApplicantIncome")

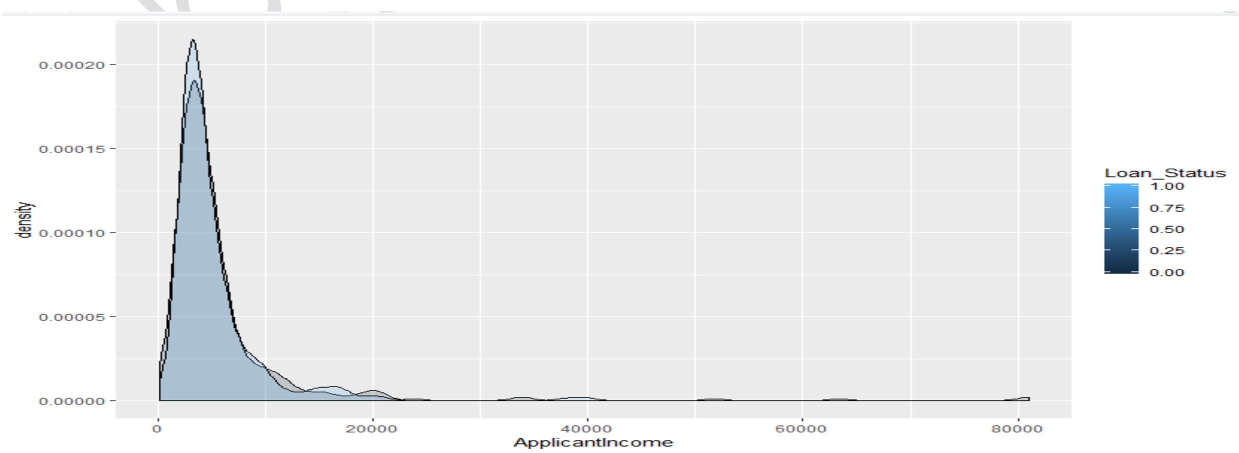
aggregate(LoanData[myvars], by=list(Loan_Status=LoanData$Loan_Status), mystats)

```
Console  Terminal ×
U:/R/R project/Final Project/
+ }
> myvars <- c("ApplicantIncome")
>
> aggregate(LoanData[myvars], by=list(Loan_Status=LoanData$Loan_Status), mystats)
  Loan_Status ApplicantIncome.length ApplicantIncome.min ApplicantIncome.max ApplicantIncome.median ApplicantIncome.mean
1           0             192.000000          150.000000        81000.000000            3833.500000          5446.078125
2           1             422.000000          210.000000        63337.000000            3812.500000          5384.068720
  ApplicantIncome.IQR ApplicantIncome.stdev ApplicantIncome.skew ApplicantIncome.kurtosis
1         2976.250000           6819.558528             7.701086                77.570916
2         2894.000000           5765.441615             5.461713                40.387414
```

#So we can easily note that Applicant Income has skewed distribution (median differs from mean)

#Density plot for Applicant Income

ggplot(LoanData, aes(x=ApplicantIncome,group=Loan_Status, fill = Loan_Status)) + geom_density(adjust=1.5, alpha = 0.2)



#From the above density plot, we can say that there are more applicants whose income is less than 20,000 rupees per month.

#Violin plot for Applicant Income

 # violin plots give you an idea of the density of the data as well as outliers, that

#Violin plots are similar to box plots,

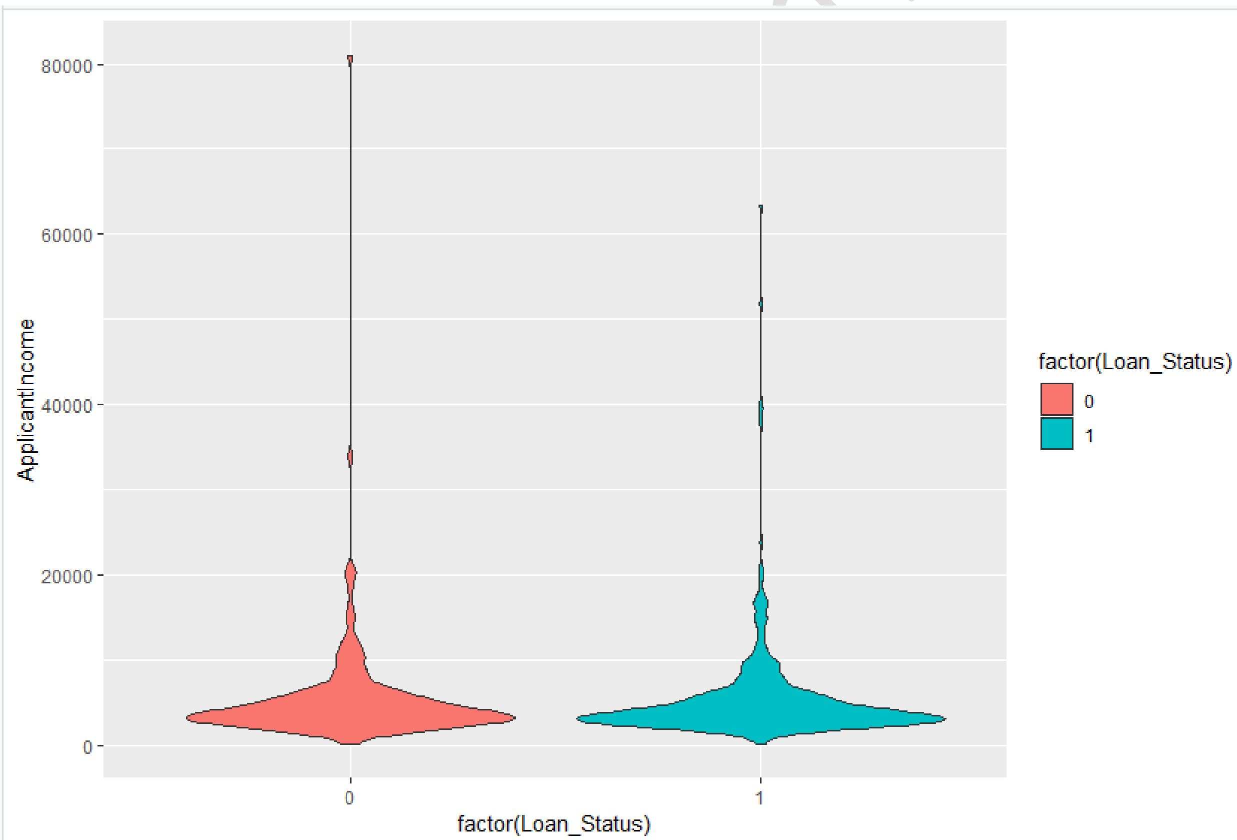
#except that they also show the probability density of the data at different values

p <- ggplot(LoanData, aes(factor(Loan_Status), ApplicantIncome))

p + geom_violin()

#Violin plot after color grading

p + geom_violin(aes(fill = factor(Loan_Status)))



#Within prepared violin plot we can note that distribution for both subgroups looks very similar.

#Both have some outliers.

#CoApplicant Income

myvars1 <- c("CoapplicantIncome")

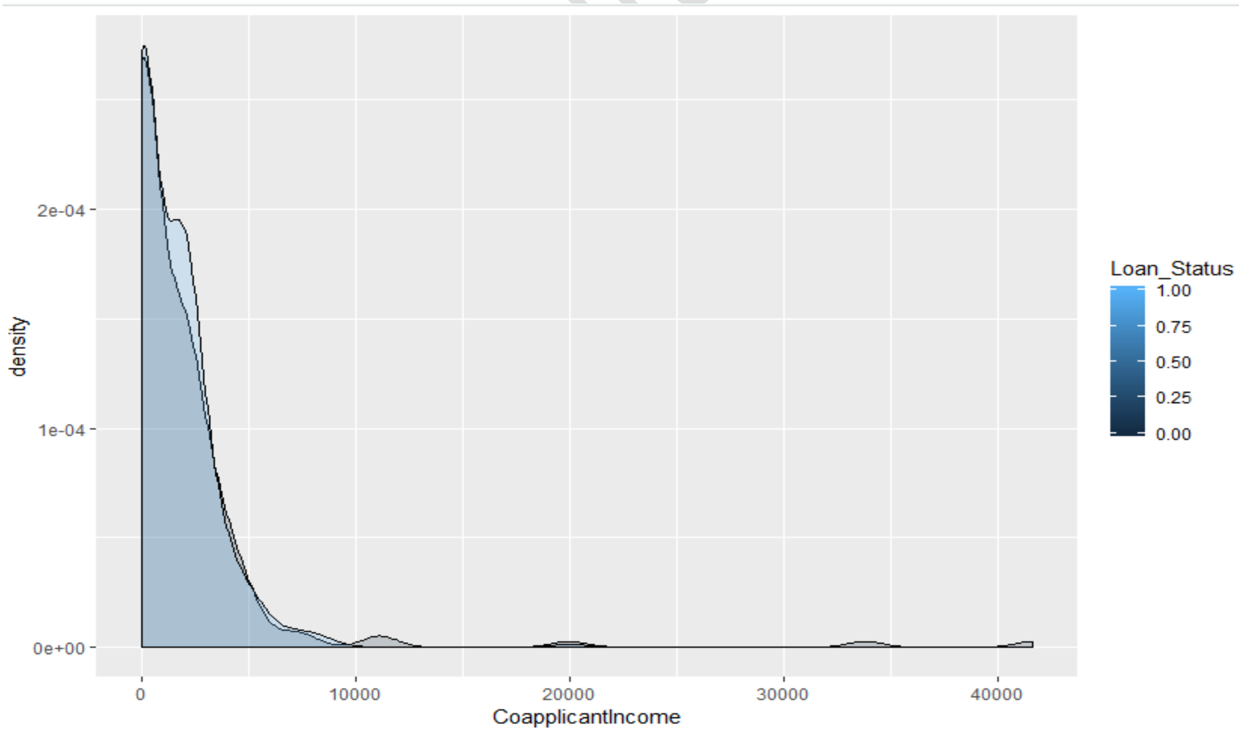
aggregate(LoanData[myvars1], by=list(Loan_Status=LoanData$Loan_Status), mystats)

```
Console   Terminal ×
U:/R/R project/Final Project/
> myvars1 <- c("CoapplicantIncome")
>
> aggregate(LoanData[myvars1], by=list(Loan_Status=LoanData$Loan_Status), mystats)
  Loan_Status CoapplicantIncome.length CoapplicantIncome.min CoapplicantIncome.max CoapplicantIncome.median
1           0               192.000000              0.000000          41667.000000               268.000000
2           1               422.000000              0.000000          20000.000000              1239.500000
  CoapplicantIncome.mean CoapplicantIncome.IQR CoapplicantIncome.stdev CoapplicantIncome.skew CoapplicantIncome.kurtosis
1            1877.807292           2273.750000             4384.060103               6.386764                  48.840880
2            1504.516398           2297.250000             1924.754855               3.019973                  20.362443
> |
```

#Subgroup of accepted loans is much more numerous. So we can easily note that Coapplicant Income has skewed distribution (median differs from mean). Very interesting is big difference between mean and median.

#Density plot for Coapplicant Income

ggplot(LoanData, aes(x=CoapplicantIncome,group=Loan_Status, fill = Loan_Status)) + geom_density(adjust=1.5, alpha = 0.2)



#From the above density plot, we can say that there are more applicants whose income is less than 10,000 rupees per month.

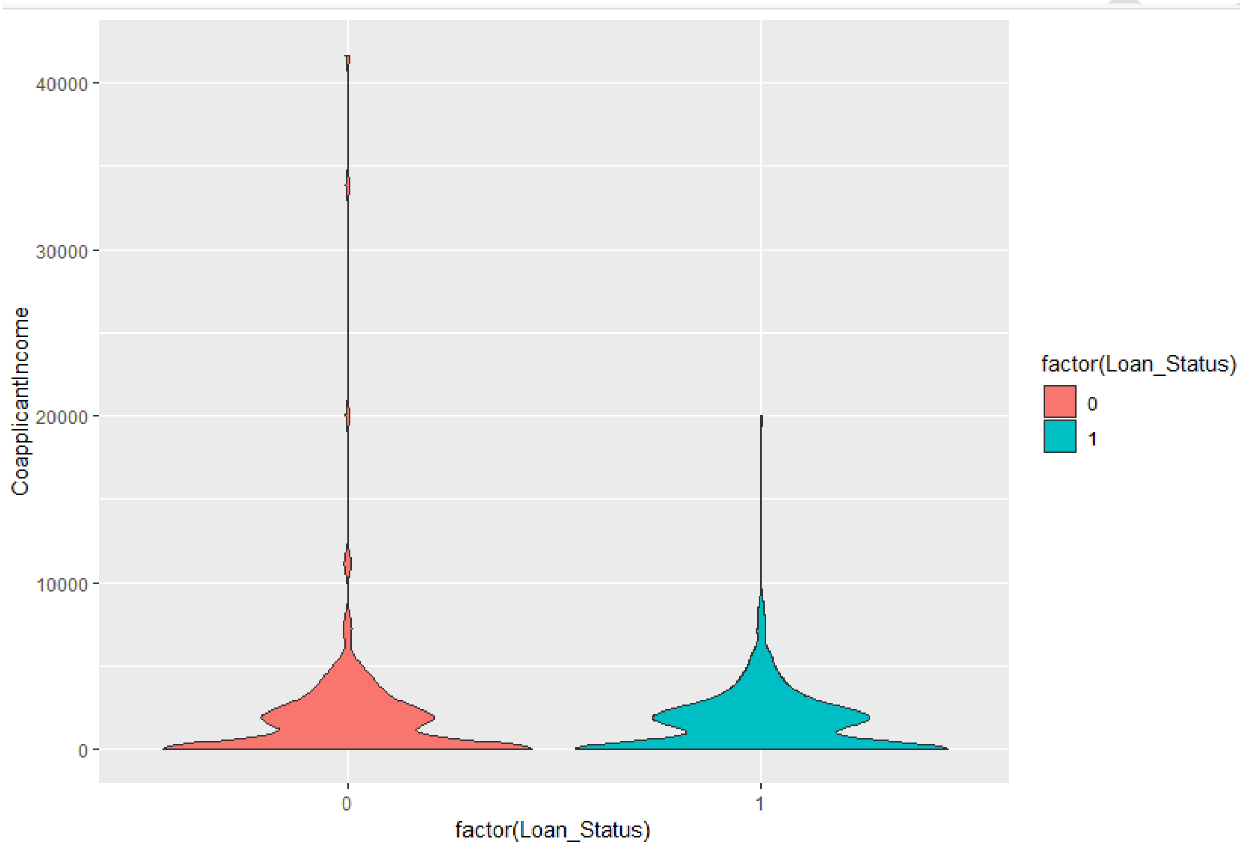
#Violin plot for Coapplicant Income

p <- ggplot(LoanData, aes(factor(Loan_Status), CoapplicantIncome))

p + geom_violin()

#Violin plot after color grading

p + geom_violin(aes(fill = factor(Loan_Status)))



#Visible is high number of coapplicants with income equal to 0.

#LoanAmount

myvars2 <- c("LoanAmount")

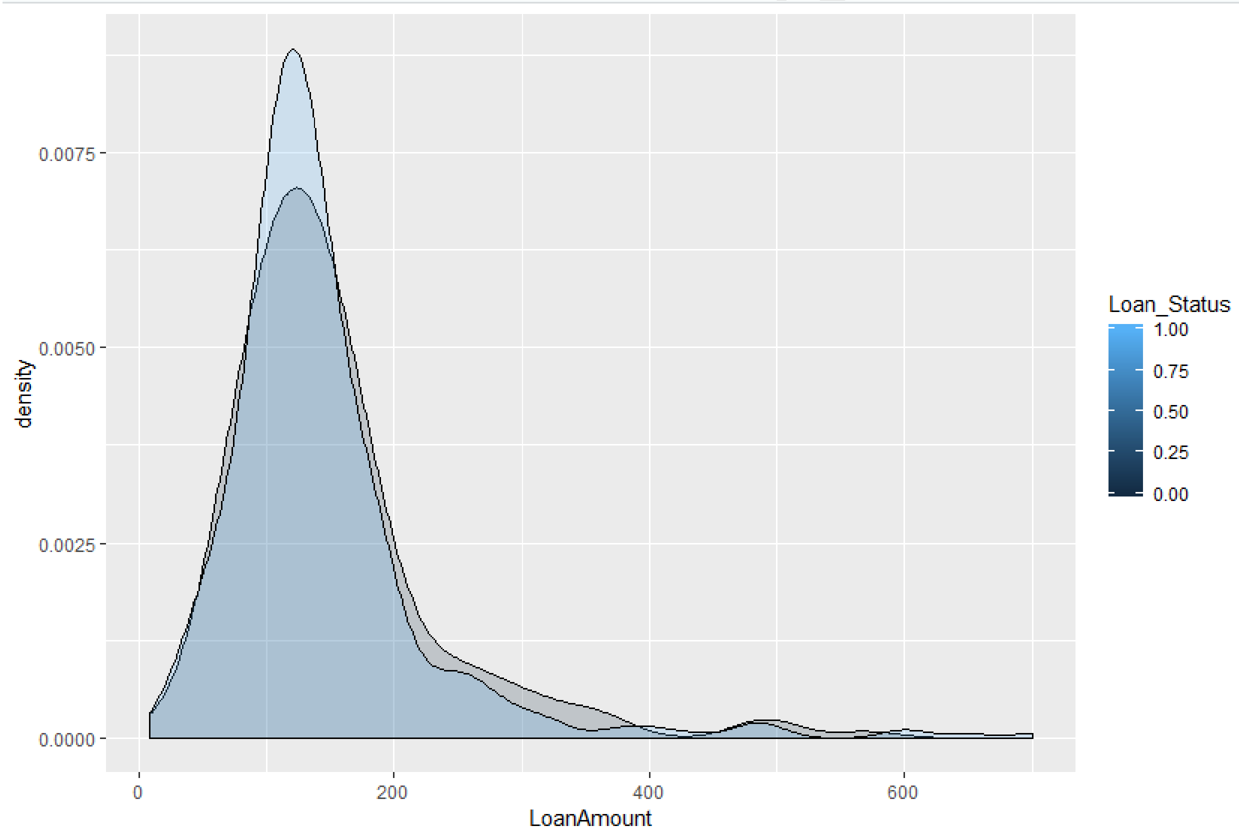
aggregate(LoanData[myvars2], by=list(Loan_Status=LoanData$Loan_Status), mystats)

```
Console   Terminal ×
U:/R/R project/Final Project/
> myvars2 <- c("LoanAmount")
>
> aggregate(LoanData[myvars2], by=list(Loan_Status=LoanData$Loan_Status), mystats)
  Loan_Status LoanAmount.length LoanAmount.min LoanAmount.max LoanAmount.median LoanAmount.mean LoanAmount.IQR
1           0        192.000000       9.000000     570.000000        133.500000      150.945488      70.250000
2           1        422.000000      17.000000     700.000000        128.000000      144.349606      60.000000
  LoanAmount.stdev LoanAmount.skew LoanAmount.kurtosis
1        83.361163        2.136255            6.256147
2        84.361109        2.966596           12.767484
> |
```

#Similarly to Coapplicant Income, accepted loans subgroups is more numerous than rejected. Median and means in both subgroups are very similar.

#Density plot for LoanAmount

ggplot(LoanData, aes(x=LoanAmount,group=Loan_Status, fill = Loan_Status)) + geom_density(adjust=1.5, alpha = 0.2)



#Maximum amount with accepted loans is greater than maximum amount within rejected #loans. We could have expected that loans with greater amount are more likely to be rejected.

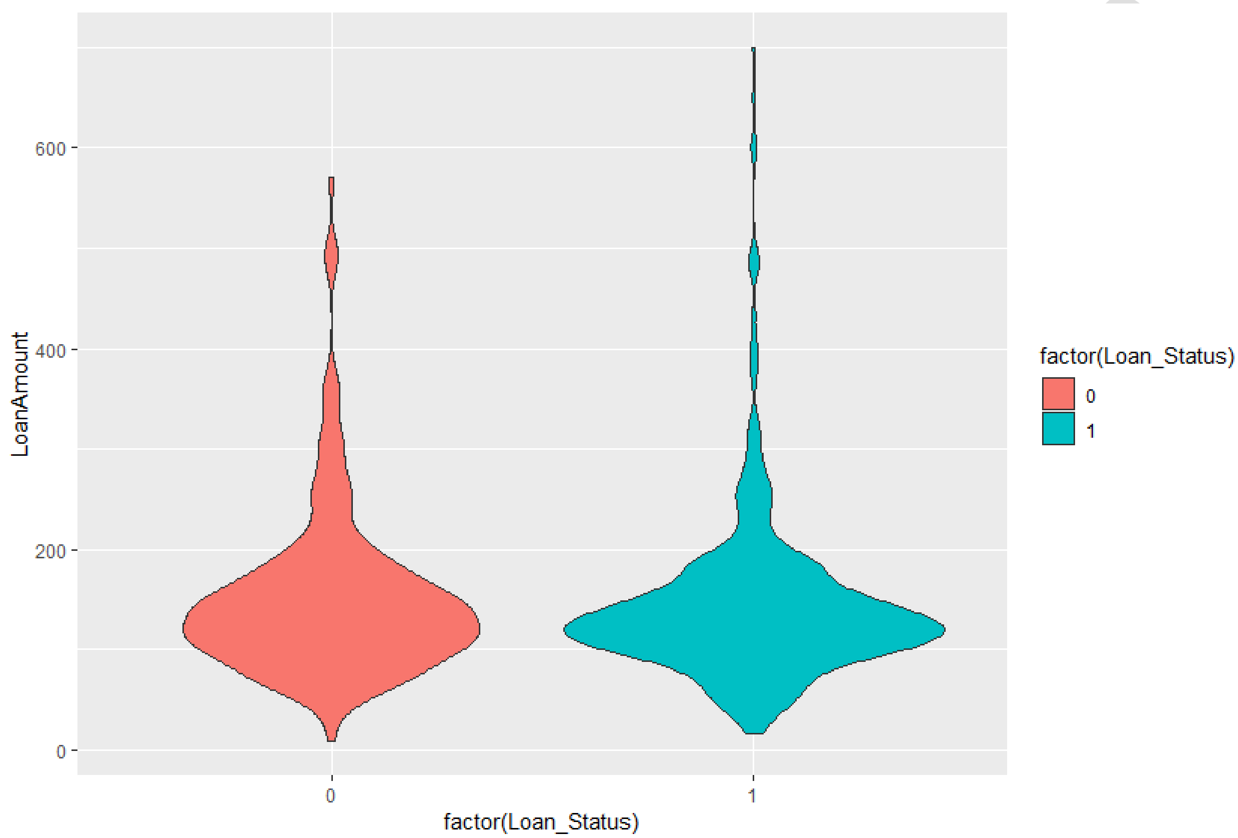
#Violin plot for LoanAmount

p <- ggplot(LoanData, aes(factor(Loan_Status), LoanAmount))

p + geom_violin()

#Violin plot after color grading

p + geom_violin(aes(fill = factor(Loan_Status)))



 # Accepted subgroup is more dense around 100 - 150. Rejected has higher IQR and both subgroups have some outliers.

#LoanAmountTerm

myvars3 <- c("Loan_Amount_Term")

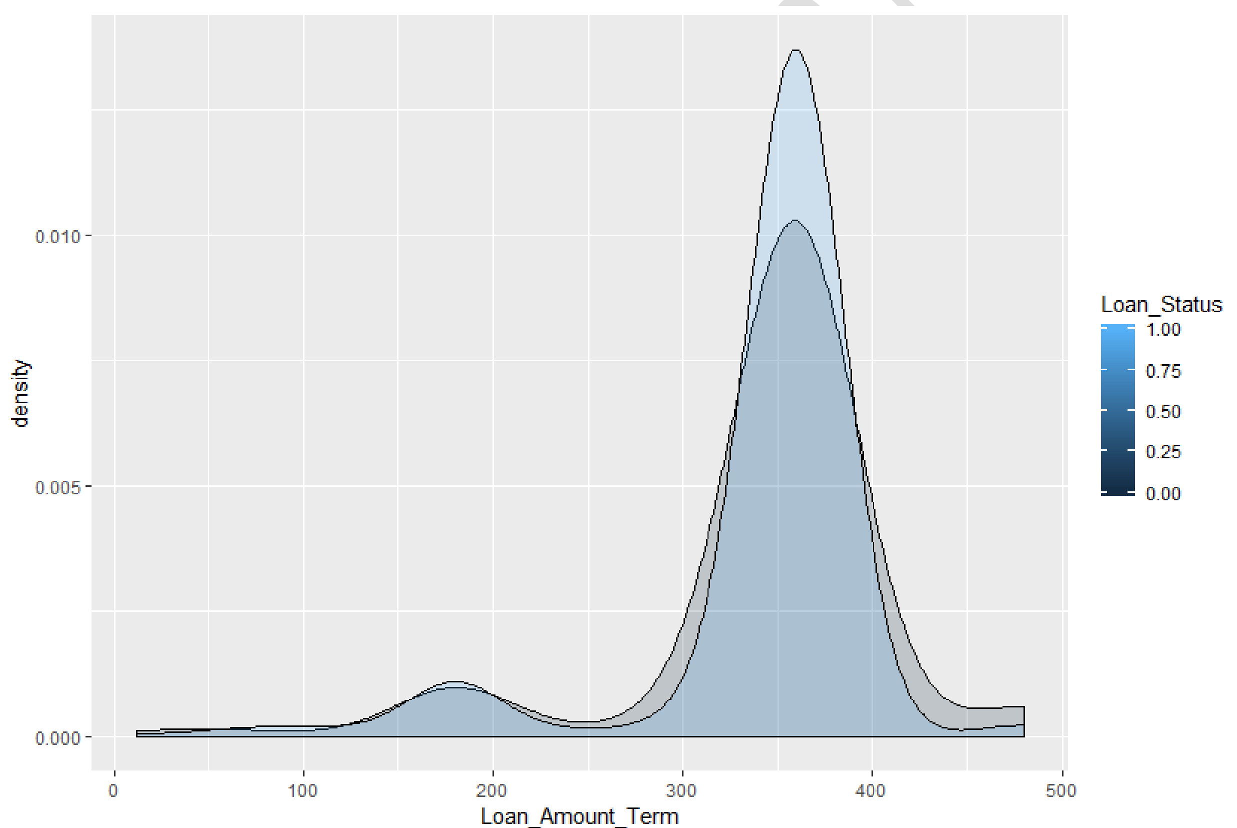
aggregate(LoanData[myvars3], by=list(Loan_Status=LoanData$Loan_Status), mystats)

```
Console  Terminal ×
U:/R/R project/Final Project/
> myvars3 <- c("Loan_Amount_Term")
>
> aggregate(LoanData[myvars3], by=list(Loan_Status=LoanData$Loan_Status), mystats)
  Loan_Status Loan_Amount_Term.length Loan_Amount_Term.min Loan_Amount_Term.max Loan_Amount_Term.median
1           0              192.000000            36.000000           480.000000              360.000000
2           1              422.000000            12.000000           480.000000              360.000000
  Loan_Amount_Term.mean Loan_Amount_Term.IQR Loan_Amount_Term.stdev Loan_Amount_Term.skew Loan_Amount_Term.kurtosis
1            344.000000             0.000000              68.143673             -1.982760                  5.851076
2            341.090047             0.000000              62.644087             -2.600987                  7.253144
>
```

#Two subgroups have very similar distributions with difference within kurtosis.

#Density plot for LoanAmount Term

ggplot(LoanData, aes(x=Loan_Amount_Term,group=Loan_Status, fill = Loan_Status)) + geom_density(adjust=1.5, alpha = 0.2)



#From the above graph we can say that there are more applicants who loan amount term lies between 350-400 months. Hence this region is densely populated.

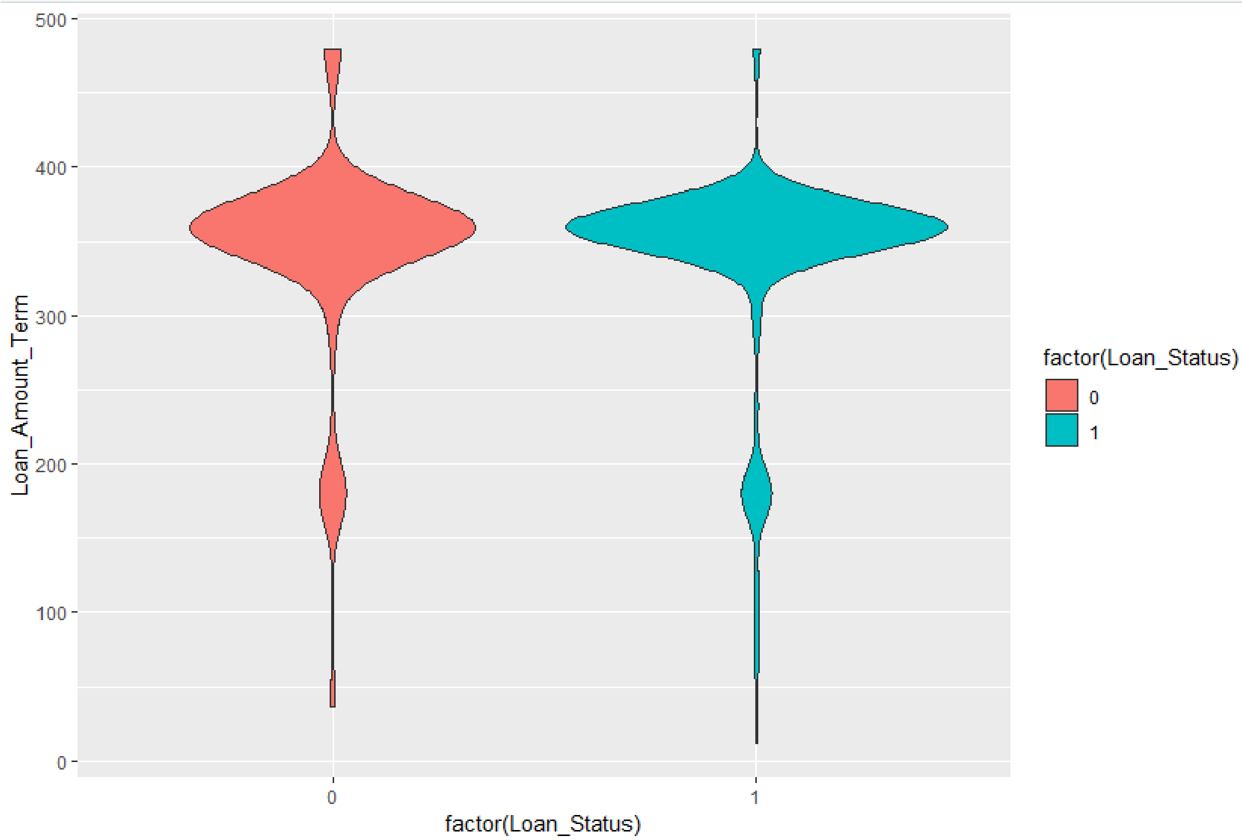
#Violin plot for LoanAmount Term

p <- ggplot(LoanData, aes(factor(Loan_Status), Loan_Amount_Term))

p + geom_violin()

#Violin plot after color grading

p + geom_violin(aes(fill = factor(Loan_Status)))



 # Minimum term of applied loans was 12 months. Maximum 480 months. Applicants usually apply for loans with term close to 30 years.

#CORRELATION FOR CONTINUOUS VARIABLES

#Now is the time to analyze the continuous variables correlation.

#It is crucial to track highly correlated variables in order to prevent multicollinearity prematurely.

 # The Pearson correlation assesses the degree of linear relationship between two quantitative variables.

#I will not use Pearson to calculate correlation coefficients as this method is highly sensitive

#to non-normal distribution and outliers presence.

#I use Kendall's tau-b coefficient instead of Pearson's,

#which is more effective in determining whether two non-parametric data samples are correlated with ties.

K = LoanData %>% select(ApplicantIncome, CoapplicantIncome, LoanAmount, Loan_Amount_Term) %>% na.omit()

K_cor = cor(K, method = "kendall")

K_cor

```
Console   Terminal ×
U:/R/R project/Final Project/
> K = LoanData %>% select(ApplicantIncome, CoapplicantIncome, LoanAmount, Loan_Amount_Term) %>% na.omit()
> K_cor = cor(K, method = "kendall")
> K_cor
                  ApplicantIncome CoapplicantIncome LoanAmount Loan_Amount_Term
ApplicantIncome        1.00000000       -0.23022170  0.3610582      -0.02583562
CoapplicantIncome     -0.23022170        1.00000000  0.1792858      -0.01979046
LoanAmount             0.36105817        0.17928579  1.0000000       0.04580490
Loan_Amount_Term      -0.02583562       -0.01979046  0.0458049       1.00000000
> |
```

col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))

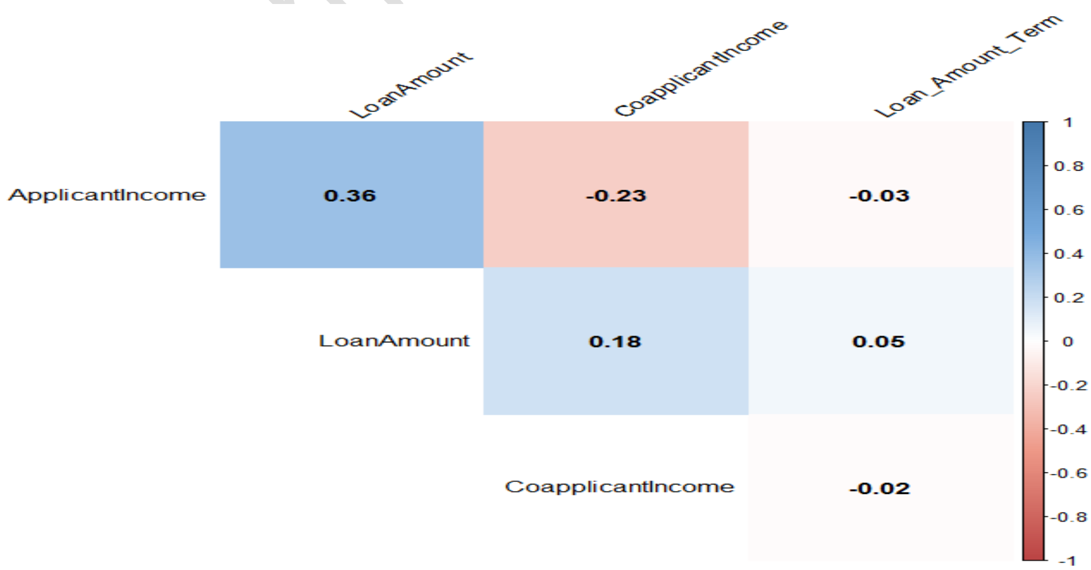
corrplot(K_cor, method="color", col=col(200),

type="upper", order="hclust",

addCoef.col = "black", # Add coefficient of correlation

tl.col="black", tl.srt=45, #Text label color and rotation

diag=FALSE)



#There are very few cont. variables so matrix is simple. We can see that:

#Applicant Income and Loan Amount are moderately correlated.

#Loan Amount and Coapplicant Income are negatively weakly associated.

#Coapplicant Income and Loan Amount are weakly associated.

#The rest is very weakly correlated.

<u>CONVERTING THE DATA TO NUMERICALS TO PERFORM MULTIPLE REGRESSION ANALYSIS</u>

```
LoanData <- read.csv("LoanData.csv",header=TRUE)
LoanData

# Structure of Loan dataframe to see if the data is structured or not
str(LoanData)

#recoding Gender for data where Male to 1 and Female to 0
Loan_reg <- LoanData%>%  mutate(Gender= ifelse(Gender == "Male",1,0))
str(Loan_reg)

#recoding Marital status for data where Married="Yes" to 1 and Married="No" to 0
Loan_reg1 <- Loan_reg %>%  mutate(Married= ifelse(Married == "Yes",1,0))
str(Loan_reg1)

#recoding Education for data where Education="Graduate" to 1 and "Not Graduate" to 0
Loan_reg2 <- Loan_reg1 %>%  mutate(Education= ifelse(Education == "Graduate",1,0))
str(Loan_reg2)

#recoding Self_Employed for data where Self_Employed="Yes" to 1 and "No" to 0
Loan_reg3 <- Loan_reg2 %>%  mutate(Self_Employed= ifelse(Self_Employed == "Yes",1,0))
str(Loan_reg3)
```

#recoding Property_Area for data where Rural=0, Urban=1 and Semiurban=2

Loan_reg3$Property_Area

Loan_reg3$Property_Area = factor(Loan_reg3$Property_Area,levels =c('Rural', 'Urban', 'Semiurban'),labels = c(0, 1, 2))

str(Loan_reg3)

#recoding Credit_History for data where Credit_History="Met" to 1 and "Unmet" to 0

Loan_reg4 <- Loan_reg3 %>% mutate(Credit_History= ifelse(Credit_History == "Met",1,0))

Loan_model <-Loan_reg4

str(Loan_model)

```
Console   Terminal ×
U:/R/R project/Final Project/
> str(Loan_model)
'data.frame':   614 obs. of  13 variables:
 $ Loan_ID          : Factor w/ 614 levels "LP001002","LP001003",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ Gender           : num  1 1 1 1 1 1 1 1 1 1 ...
 $ Married          : num  0 1 1 1 0 1 1 1 1 1 ...
 $ Dependents       : int  0 1 0 0 0 2 0 3 2 1 ...
 $ Education        : num  1 1 1 0 1 1 0 1 1 1 ...
 $ Self_Employed    : num  0 0 1 0 0 1 0 0 0 0 ...
 $ ApplicantIncome  : int  5849 4583 3000 2583 6000 5417 2333 3036 4006 12841 ...
 $ CoapplicantIncome: num  0 1508 0 2358 0 ...
 $ LoanAmount       : num  146 128 66 120 141 ...
 $ Loan_Amount_Term : int  360 360 360 360 360 360 360 360 360 360 ...
 $ Credit_History   : num  1 1 1 1 1 1 1 0 1 1 ...
 $ Property_Area    : Factor w/ 3 levels "0","1","2": 2 1 2 2 2 2 2 3 2 3 ...
 $ Loan_Status      : int  1 0 1 1 1 1 1 0 1 0 ...
>
```

#save the file in our current working directory

write.table(Loan_model,file="Loan_model.csv",row.names=F,sep=",")

#Sample Loan_model data

Sample Dataset After conversion:

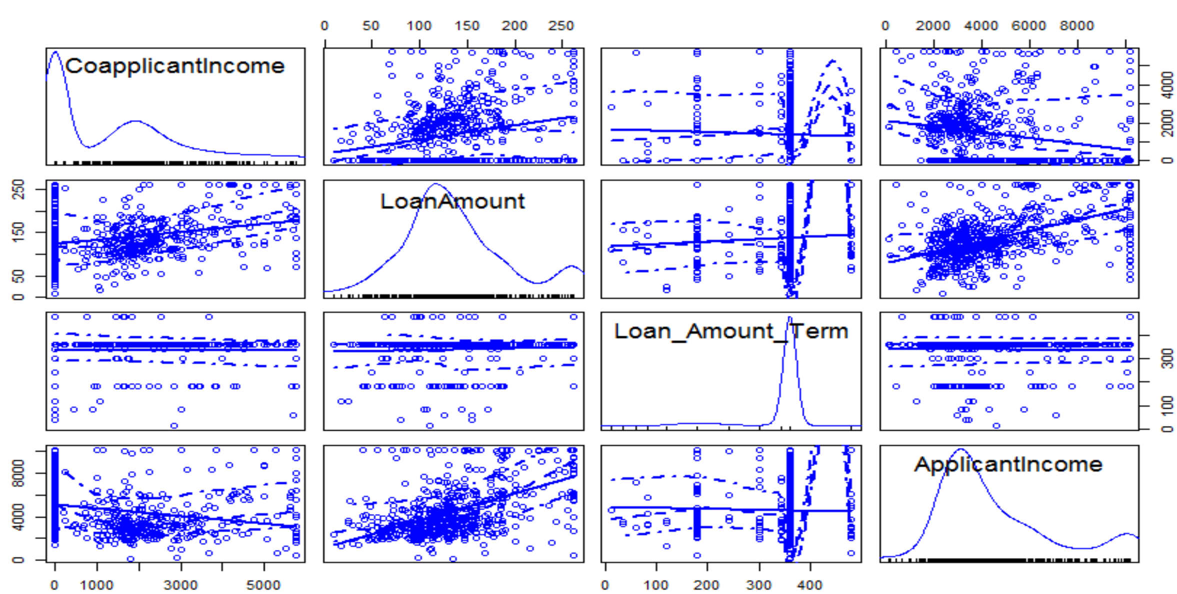
	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
1	LP001002	1	0	0	1	0	5849	0	146.4122	360	1	1	1
2	LP001003	1	1	1	1	0	4583	1508	128.0000	360	1	0	0
3	LP001005	1	1	0	1	1	3000	0	66.0000	360	1	1	1
4	LP001006	1	1	0	0	0	2583	2358	120.0000	360	1	1	1
5	LP001008	1	0	0	1	0	6000	0	141.0000	360	1	1	1
6	LP001011	1	1	2	1	1	5417	4196	267.0000	360	1	1	1
7	LP001013	1	1	0	0	0	2333	1516	95.0000	360	1	1	1
8	LP001014	1	1	3	1	0	3036	2504	158.0000	360	0	2	0
9	LP001018	1	1	2	1	0	4006	1526	168.0000	360	1	1	1
10	LP001020	1	1	1	1	0	12841	10968	349.0000	360	1	2	0
11	LP001024	1	1	2	1	0	3200	700	70.0000	360	1	1	1
12	LP001027	1	1	2	1	0	2500	1840	109.0000	360	1	1	1
13	LP001028	1	1	2	1	0	3073	8106	200.0000	360	1	1	1
14	LP001029	1	0	0	1	0	1853	2840	114.0000	360	1	0	0
15	LP001030	1	1	2	1	0	1299	1086	17.0000	120	1	1	1
16	LP001032	1	0	0	1	0	4950	0	125.0000	360	1	1	1
17	LP001034	1	0	1	0	0	3596	0	100.0000	240	0	1	1
18	LP001036	0	0	0	1	0	3510	0	76.0000	360	0	1	0
19	LP001038	1	1	0	0	0	4887	0	133.0000	360	1	0	0
20	LP001041	1	1	0	1	0	2600	3500	115.0000	342	1	1	1
21	LP001043	1	1	0	0	0	7660	0	104.0000	360	0	1	0
22	LP001046	1	1	1	1	0	5955	5625	315.0000	360	1	1	1
23	LP001047	1	1	0	0	0	2600	1911	116.0000	360	0	2	0
24	LP001050	0	1	2	0	0	3365	1917	112.0000	360	0	0	0
25	LP001052	1	1	1	1	0	3717	2925	151.0000	360	0	2	0

SCATTERPLOT FOR CONTINUOUS VARIABLES

#When you need to look at several plots, such as at the beginning of a multiple regression analysis,

#a scatter plot matrix is a very useful tool.

scatterplotMatrix(formula=~CoapplicantIncome+LoanAmount+Loan_Amount_Term+ApplicantIncome, data=Loan_model, diagonal="histogram")



#As seen in the Violin and scatter plots the ApplicantIncome,CoapplicantIncome and LoanAmount has outliers

 # and we are treating these factors to improve the performance

OUTLIER TREATMENT

Outlier Treatment for ApplicantIncome

*bench <- 5795 + 1.5*IQR(Loan_model$ApplicantIncome) #Q3 + 1.5*IQR(data$Age)*

bench

```
Console   Terminal ×
U:/R/R project/Final Project/
> bench <- 5795 + 1.5*IQR(Loan_model$ApplicantIncome) #Q3 + 1.5*IQR(data$Age)
> bench
[1] 10171.25
>
```

#WINsORIZING method of treating outlier

Loan_model$ApplicantIncome[Loan_model$ApplicantIncome > bench]

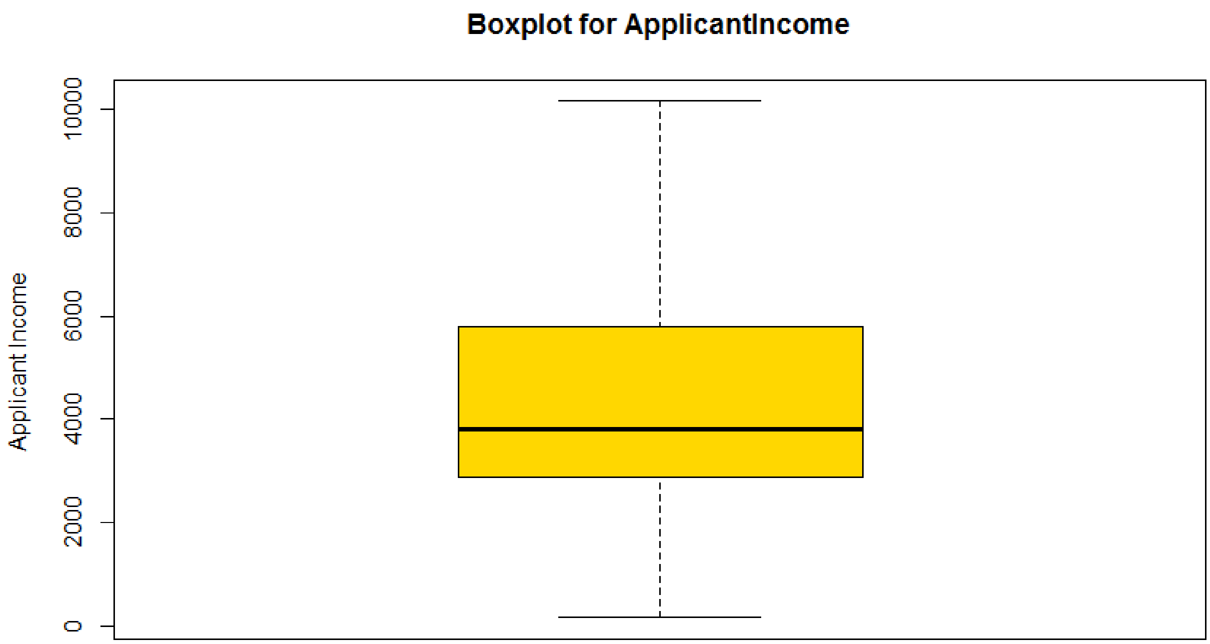
```
Console   Terminal ×
U:/R/R project/Final Project/
> Loan_model$ApplicantIncome[Loan_model$ApplicantIncome > bench]
 [1] 12841 12500 11500 10750 13650 11417 14583 10408 23803 10513 20166 14999 11757 14866 39999 51763 33846 39147 12000
[20] 11000 16250 14683 11146 14583 20667 20233 15000 63337 19730 15759 81000 14880 12876 10416 37719 16692 16525 16667
[39] 10833 18333 17263 20833 13262 17500 11250 18165 19484 16666 16120 12000
>
```

Loan_model$ApplicantIncome[Loan_model$ApplicantIncome > bench] <- bench

summary(Loan_model$ApplicantIncome)

```
Console   Terminal ×
U:/R/R project/Final Project/
> summary(Loan_model$ApplicantIncome)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    150    2878    3812    4617    5795   10171
>
```

boxplot(Loan_model$ApplicantIncome, main = "Boxplot for ApplicantIncome",ylab="Applicant Income ",col=(c("gold")))



#Outlier Treatment for CoapplicantIncome

*bench1 <- 2297 + 1.5*IQR(Loan_model$CoapplicantIncome) #Q3 + 1.5*IQR(data$Age)*

bench1

```
Console  Terminal
U:/R/R project/Final Project/
> bench1 <- 2297 + 1.5*IQR(Loan_model$CoapplicantIncome) #Q3 + 1.5*IQR(data$Age)
> bench1
[1] 5742.875
>
```

#WINsORIZING method of treating outlier

Loan_model$CoapplicantIncome[Loan_model$CoapplicantIncome > bench1]

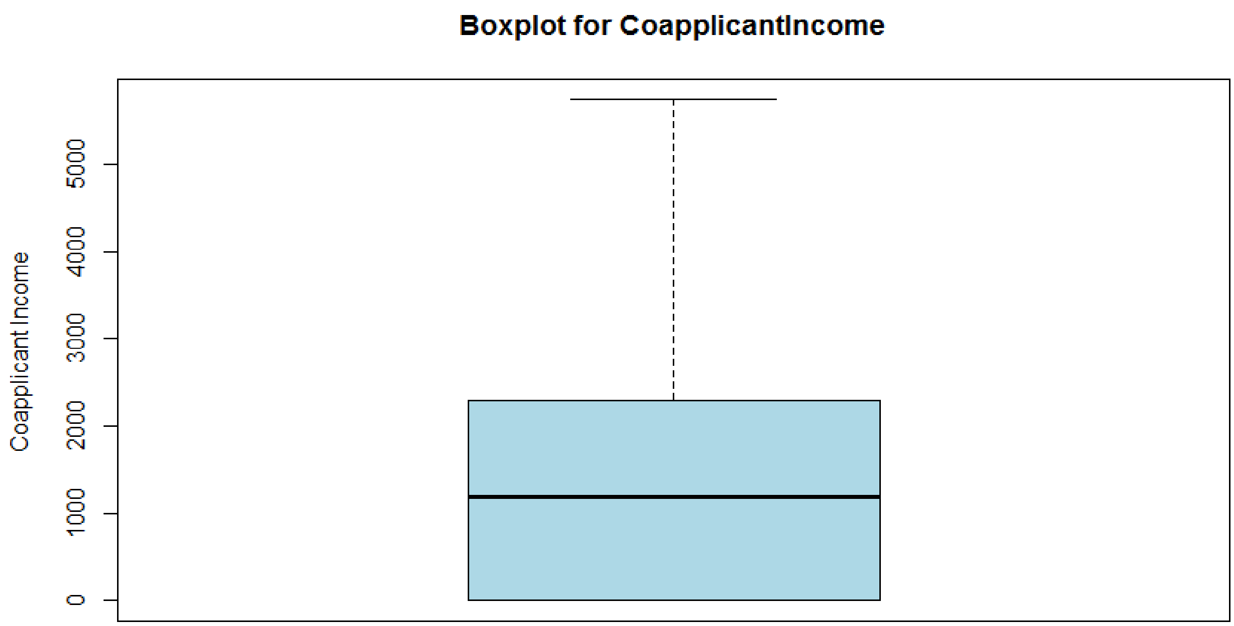
```
Console  Terminal
U:/R/R project/Final Project/
> Loan_model$CoapplicantIncome[Loan_model$CoapplicantIncome > bench1]
 [1] 10968  8106  7210  8980  7750 11300  7250  7101  6250  7873 20000 20000  8333  6667  6666  7166 33837 41667
>
```

Loan_model$CoapplicantIncome[Loan_model$CoapplicantIncome > bench1] <- bench1

summary(Loan_model$CoapplicantIncome)

```
Console   Terminal ×
U:/R/R project/Final Project/
> summary(Loan_model$CoapplicantIncome)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
      0       0    1188    1420    2297    5743 
> 
```

boxplot(Loan_model$CoapplicantIncome, main = "Boxplot for CoapplicantIncome",ylab="Coapplicant Income ",col=(c("lightblue")))



#Outlier Treatment for LoanAmount

*bench2 <- 164.8 + 1.5*IQR(Loan_model$LoanAmount) #Q3 + 1.5*IQR(data$Age)*

bench2

```
Console   Terminal ×
U:/R/R project/Final Project/
> bench2 <- 164.8 + 1.5*IQR(Loan_model$LoanAmount) #Q3 + 1.5*IQR(data$Age)
> bench2
[1] 261.55
> 
```

#WINsORIZING method of treating outlier

Loan_model$LoanAmount [Loan_model$LoanAmount > bench2]

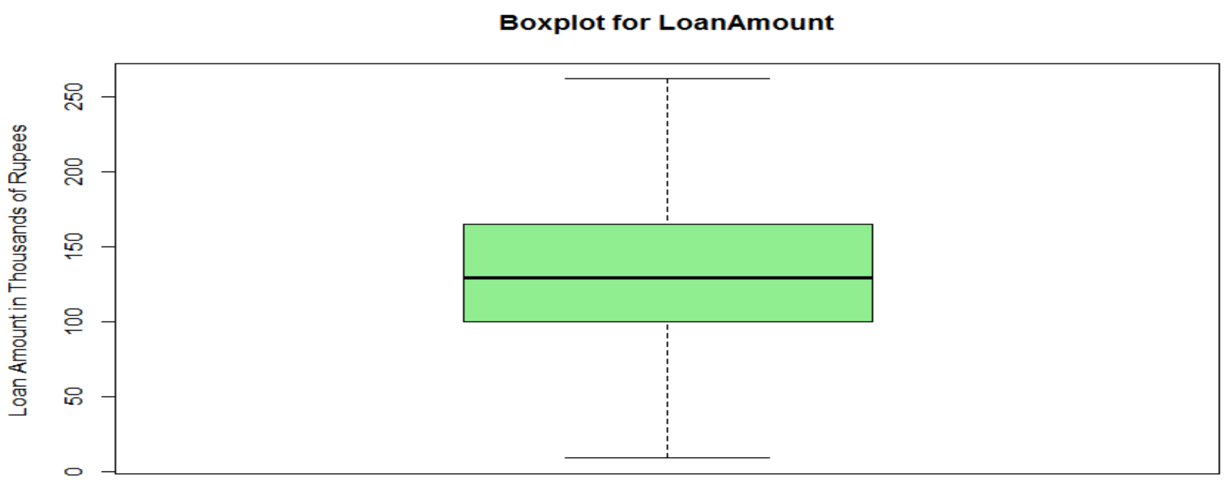
```
Console  Terminal ×
U:/R/R project/Final Project/
> Loan_model$LoanAmount [Loan_model$LoanAmount  > bench2]
 [1] 267 349 315 320 286 312 265 370 650 290 600 275 700 495 280 279 304 330 436 480 300 376 490 308 570 380 296 275 360
[30] 405 500 480 311 480 400 324 600 275 292 350 496
>
```

Loan_model$LoanAmount [Loan_model$LoanAmount > bench2] <- bench2

summary(Loan_model$LoanAmount)

```
Console  Terminal ×
U:/R/R project/Final Project/
> summary(Loan_model$LoanAmount )
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    9.0   100.2   129.0   138.0   164.8   261.6 
>
```

boxplot(Loan_model$LoanAmount, main = "Boxplot for LoanAmount",ylab="Coapplicant Income ",col=(c("lightgreen")))



#The outliers have all been treated and the data is now clean to an appreciable level.

MULTIPLE REGRESSION ANALYSIS

 # performing Multiple linear regression between Loan_Status and all variables

#to evaluate the model performance.

Loan_pef <- lm(Loan_Status ~ Gender+Married+Dependents+Education+Self_Employed+ApplicantIncome+CoapplicantIncome+

LoanAmount+Loan_Amount_Term+Credit_History+Property_Area,

data = Loan_model)

summary(Loan_pef)

```
Console   Terminal ×
U:/R/R project/Final Project/
> summary(Loan_pef)

Call:
lm(formula = Loan_Status ~ Gender + Married + Dependents + Education +
    Self_Employed + ApplicantIncome + CoapplicantIncome + LoanAmount +
    Loan_Amount_Term + Credit_History + Property_Area, data = Loan_model)

Residuals:
    Min      1Q  Median      3Q     Max
-1.0187 -0.2911  0.1528  0.2396  0.8449

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        2.520e-01  1.172e-01   2.150  0.03195 *
Gender            -7.952e-03  4.516e-02  -0.176  0.86030
Married            9.908e-02  4.024e-02   2.463  0.01408 *
Dependents        -5.646e-04  1.808e-02  -0.031  0.97510
Education          6.176e-02  4.232e-02   1.459  0.14500
Self_Employed      3.078e-03  5.079e-02   0.061  0.95170
ApplicantIncome    4.988e-06  9.639e-06   0.517  0.60500
CoapplicantIncome  1.284e-05  1.261e-05   1.018  0.30930
LoanAmount        -7.681e-04  4.240e-04  -1.811  0.07057 .
Loan_Amount_Term  -9.838e-05  2.670e-04  -0.368  0.71265
Credit_History     4.698e-01  4.037e-02  11.636  < 2e-16 ***
Property_Area1     4.364e-02  4.304e-02   1.014  0.31095
Property_Area2     1.312e-01  4.165e-02   3.151  0.00171 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4141 on 601 degrees of freedom
Multiple R-squared:  0.219,	Adjusted R-squared:  0.2034
F-statistic: 14.05 on 12 and 601 DF,  p-value: < 2.2e-16

> |
```

#The summary statistics above tells us a number of things.

#We can consider a linear model to be statistically significant only when these p-Values are less.

#Higher the t-value, the better the model is.

#The t-statistic is the coefficient estimate divided by the standard error. If your regression is based on what

#statisticians call a "large" sample (30 or more observations), a t-statistic greater than 2 (or less than -2)

#indicates the coefficient is significant with >95% confidence.

#A predictor that has a low p-value is likely to be a meaningful addition to your model

#because changes in the predictor's value are related to changes in the response variable.

#Conversely, a larger (insignificant) p-value suggests that changes in the predictor are not associated with changes in the response.

#From our summary we see that p-value of 'Self_Employed' and 'Dependents' is high and t-value is low so we will try eliminating that variables

#and see if our model accuracy is improved or not.

#Residual Standard error is 0.4141 that is deviation from getting perfect linear regression.

#R-squared is a statistical measure of how close the data are to the fitted regression line.

#It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.

#The definition of R-squared is fairly straight-forward; it is the percentage of the response variable variation that is explained by a linear model.

#R^2 and Adj R^2 gives accuracy of model, we will consider Adj R^2 to be more accurate as R^2 changes with added variables.

#In general, the higher the R-squared, the better the model fits your data.

#Our model accuracy at this point is 20.34%.

#The F-test of the overall significance is a specific form of the F-test.

#F-value gives overall performance of the model that is 14.05.

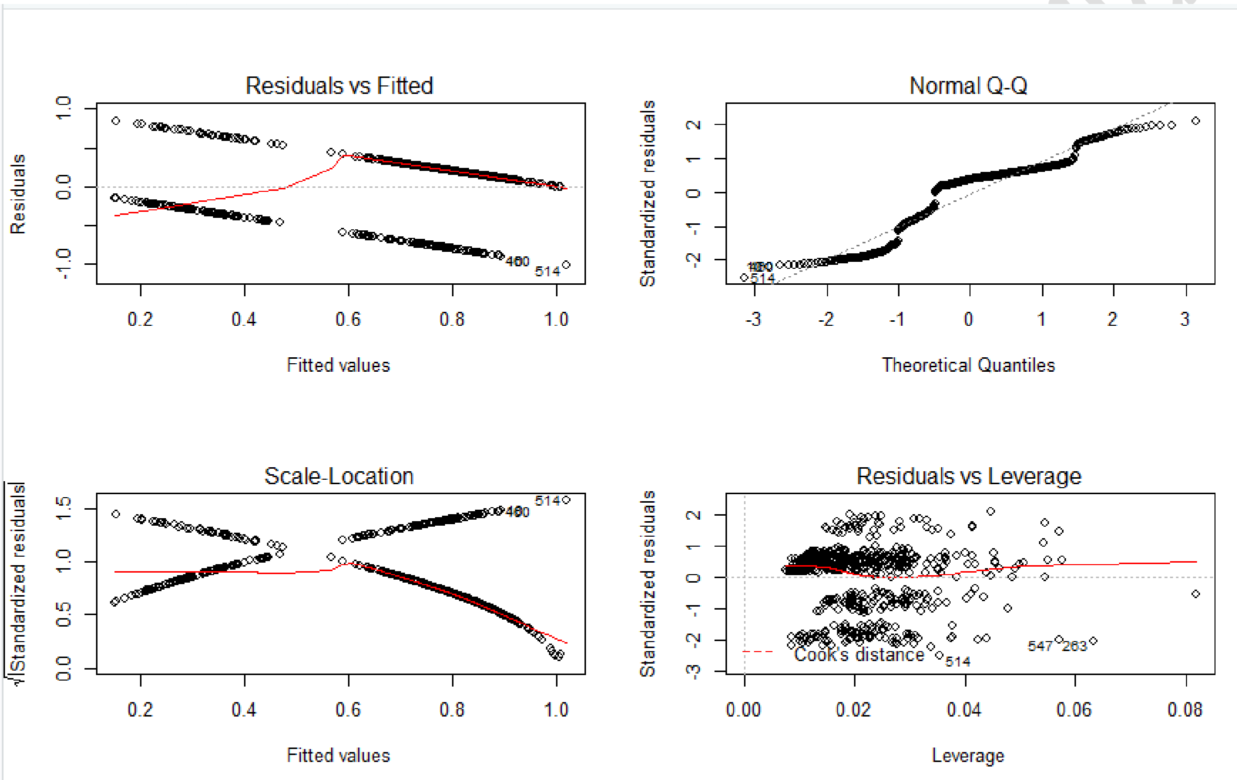
 # Simple Regression Diagnostics

 # one way to evaluate the statistical assumptions in regression analysis

 # is to plot the results of lm

par(mfrow = c(2, 2))

plot(Loan_pef)



#Removing Self_Employed and Dependents variables

Loan_pef1 <- lm(Loan_Status ~ Gender+Married+Education+ApplicantIncome+ CoapplicantIncome+LoanAmount+Loan_Amount_Term+Credit_History+Property_Area,

data = Loan_model)

summary(Loan_pef1)

```
Console   Terminal ×
U:/R/R project/Final Project/
> summary(Loan_pef1)

Call:
lm(formula = Loan_Status ~ Gender + Married + Education + ApplicantIncome +
    CoapplicantIncome + LoanAmount + Loan_Amount_Term + Credit_History +
    Property_Area, data = Loan_model)

Residuals:
    Min      1Q  Median      3Q     Max
-1.0188 -0.2901  0.1527  0.2397  0.8475

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        2.520e-01  1.166e-01   2.162  0.03104 *
Gender            -8.143e-03  4.499e-02  -0.181  0.85645
Married            9.873e-02  3.822e-02   2.583  0.01002 *
Education          6.174e-02  4.214e-02   1.465  0.14342
ApplicantIncome    5.113e-06  9.414e-06   0.543  0.58726
CoapplicantIncome  1.293e-05  1.245e-05   1.038  0.29960
LoanAmount        -7.707e-04  4.194e-04  -1.838  0.06659 .
Loan_Amount_Term  -9.809e-05  2.656e-04  -0.369  0.71202
Credit_History     4.697e-01  4.030e-02  11.657  < 2e-16 ***
Property_Area1     4.359e-02  4.296e-02   1.015  0.31065
Property_Area2     1.312e-01  4.158e-02   3.156  0.00168 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4134 on 603 degrees of freedom
Multiple R-squared:  0.219,	Adjusted R-squared:  0.2061
F-statistic: 16.91 on 10 and 603 DF,  p-value: < 2.2e-16

>
```

#By removing Self_Employed and Dependents our model accuracy(Adj R^2) has increased to 20.61% from 20.34%.

#Residual Standard error has also reduced from 0.4141 to 0.4134.

#F-value (higher the better) increased to 16.91 from 14.05.

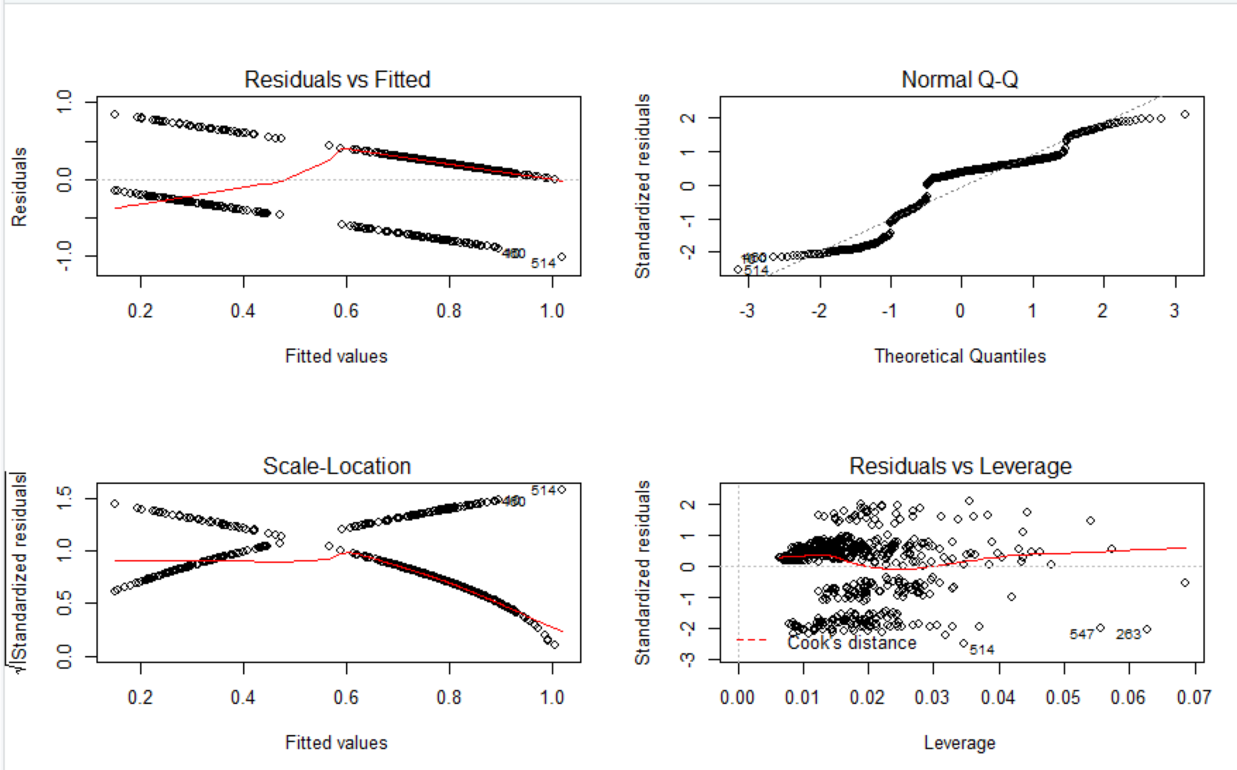
#we can see from above results that, p-value for Gender is very high

#so we will remove that variables from our model in next step and see if it improves our model.

Simple Regression Diagnostics after Removing Self_Employed and Dependents variables

par(mfrow = c(2, 2))

plot(Loan_pef1)



#Removing Gender and Loan_Amount_Term variables

Loan_pef2 <- lm(Loan_Status ~ Married+Education+ApplicantIncome+CoapplicantIncome+

LoanAmount+Credit_History+Property_Area,

data = Loan_model)

summary(Loan_pef2)

```
Console   Terminal ×
U:/R/R project/Final Project/
> summary(Loan_pef2)

Call:
lm(formula = Loan_Status ~ Married + Education + ApplicantIncome + 
    CoapplicantIncome + LoanAmount + Credit_History + Property_Area, 
    data = Loan_model)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.0051 -0.2935  0.1491  0.2434  0.8320 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)        2.137e-01  6.751e-02   3.166  0.00162 ** 
Married            9.805e-02  3.616e-02   2.711  0.00689 ** 
Education          6.106e-02  4.179e-02   1.461  0.14452    
ApplicantIncome    5.420e-06  9.313e-06   0.582  0.56078    
CoapplicantIncome  1.303e-05  1.229e-05   1.060  0.28936    
LoanAmount        -7.909e-04  4.151e-04  -1.905  0.05719 .  
Credit_History     4.693e-01  4.016e-02  11.686  < 2e-16 ***
Property_Area1     4.490e-02  4.275e-02   1.050  0.29398    
Property_Area2     1.320e-01  4.131e-02   3.194  0.00147 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4128 on 605 degrees of freedom
Multiple R-squared:  0.2188,	Adjusted R-squared:  0.2085 
F-statistic: 21.18 on 8 and 605 DF,  p-value: < 2.2e-16

> |
```

#By removing gender and Loan_Amount_Term variables our model accuracy(Adj R^2) has increased to 20.85% from 20.74%.

#Residual Standard error has also reduced from 0.4131 to 0.4128.

#F-value (higher the better) increased to 21.18 from 18.82.

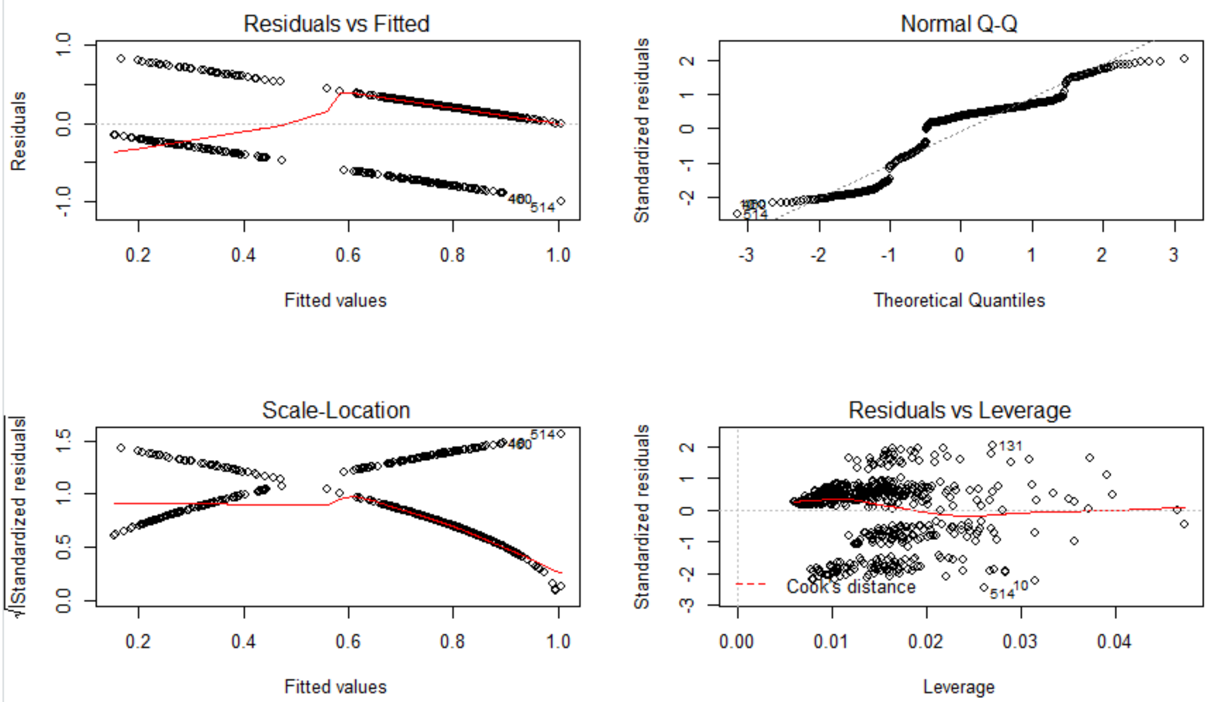
#We can see from above results that, p-value for ApplicantIncome is moderately high

#so we will remove that variable from our model in next step and see if it improves our model.

Simple Regression Diagnostics after Removing Gender and Loan_Amount_Term variables

par(mfrow = c(2, 2))

plot(Loan_pef2)



#Removing ApplicantIncome and Coapplicant Income variables

Loan_pef3 <- lm(Loan_Status ~ Married+Education+

LoanAmount+Credit_History+Property_Area,

data = Loan_model)

summary(Loan_pef3)

```
Console   Terminal ×
U:/R/R project/Final Project/
> summary(Loan_pef3)

Call:
lm(formula = Loan_Status ~ Married + Education + LoanAmount +
    Credit_History + Property_Area, data = Loan_model)

Residuals:
    Min      1Q  Median      3Q     Max
-0.9496 -0.2876  0.1469  0.2423  0.8204

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.2222284  0.0667710   3.328 0.000927 ***
Married         0.1042197  0.0354559   2.939 0.003413 **
Education       0.0655541  0.0412929   1.588 0.112911
LoanAmount     -0.0005733  0.0003097  -1.851 0.064635 .
Credit_History  0.4689580  0.0400101  11.721  < 2e-16 ***
Property_Area1  0.0417506  0.0426116   0.980 0.327578
Property_Area2  0.1287707  0.0411665   3.128 0.001844 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4125 on 607 degrees of freedom
Multiple R-squared:  0.2174,    Adjusted R-squared:  0.2096
F-statistic:  28.1 on 6 and 607 DF,  p-value: < 2.2e-16
```

#By removing ApplicantIncome and Coapplicant Income variables our model accuracy(Adj R^2) has increased to 20.96% from 20.85%.

#Residual Standard error has also reduced from 0.4128 to 0.4125.

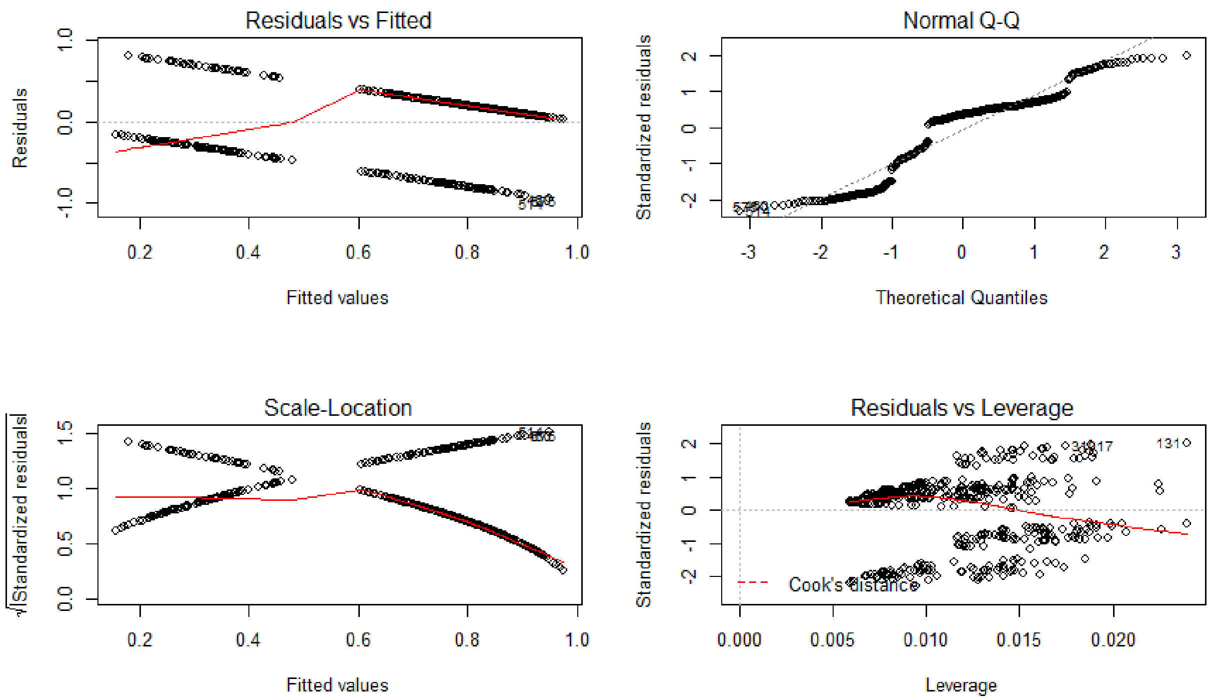
#F-value (higher the better) increased to 28.1 from 21.18.

We have improved accuracy of our model from 20.34% to 20.96%., with reducing the error rate and increasing the overall performance (F-statistic) of the model. It is a good practice to bring error rate to 0 and our model has its low error value. And accuracy depends on the data we take and always cannot get high accuracy when we study behavioral data. We see that Loan Status has a strong relation to Credit History, Married, Property area, Loan Amount and Education.

Simple Regression Diagnostics after Removing ApplicantIncome and Coapplicant Income variables

par(mfrow = c(2, 2))

plot(Loan_pef3)



#Evaluating multi-collinearity

 # Multi collinearity can be detected using the variance inflation factor (VIF).

 # For any predictor variable, the square root of the VIF indicates the

 # degree to which the confidence interval for that variables regression parameter is expanded

 # relative to a model with uncorrelated predictors.

 # VIF values are provided by the vif() function in the car package.

 # As a general rule, a sqrt >2

 # indicates a multicollinearity problem.

vif(Loan_pef3)

sqrt(vif(Loan_pef3)) > 2

```
Console   Terminal ×
U:/R/R project/Final Project/
> vif(Loan_pef3)
                   GVIF Df GVIF^(1/(2*Df))
Married        1.034476  1        1.017092
Education      1.049793  1        1.024594
LoanAmount     1.075181  1        1.036909
Credit_History 1.011707  1        1.005836
Property_Area  1.017006  2        1.004225
> sqrt(vif(Loan_pef3)) > 2
                GVIF    Df GVIF^(1/(2*Df))
Married        FALSE FALSE           FALSE
Education      FALSE FALSE           FALSE
LoanAmount     FALSE FALSE           FALSE
Credit_History FALSE FALSE           FALSE
Property_Area  FALSE FALSE           FALSE
> 
```

#No multi collinearity here

BACKWARD STEPWISE SELECTION:

 # The stepAIC() function in the MASS package performs

 # stepwise model selection (forward, backward, stepwise) using the AIC critera

 # The Akaike Information Criterion (AIC) is a method for comparing

 # models. The index takes into account a model's statistical fit and the number of

 # parameters needed to achieve this fit.

 # Models with smaller AIC values indicating adequate fit with fewer parameters are preferred.

library(MASS)

Loan_backward<- lm(Loan_Status ~ Gender+Married+Dependents+Education+Self_Employed+ApplicantIncome+CoapplicantIncome+

LoanAmount+Loan_Amount_Term+Credit_History+Property_Area,

data = Loan_model)

 # backward direction

stepAIC(Loan_backward, direction = "backward")

Console Terminal

U:/R/R project/Final Project/

```
> # backward direction
> stepAIC(Loan_backward, direction = "backward")
Start:  AIC=-1069.82
Loan_Status ~ Gender + Married + Dependents + Education + Self_Employed +
    ApplicantIncome + CoapplicantIncome + LoanAmount + Loan_Amount_Term +
    Credit_History + Property_Area

                    Df Sum of Sq    RSS     AIC
- Dependents         1    0.0002 103.06 -1071.81
- Self_Employed      1    0.0006 103.06 -1071.81
- Gender             1    0.0053 103.06 -1071.78
- Loan_Amount_Term   1    0.0233 103.08 -1071.68
- ApplicantIncome    1    0.0459 103.10 -1071.54
- CoapplicantIncome  1    0.1775 103.23 -1070.76
<none>                          103.06 -1069.82
- Education          1    0.3652 103.42 -1069.64
- LoanAmount         1    0.5627 103.62 -1068.47
- Married            1    1.0398 104.10 -1065.65
- Property_Area      2    1.8265 104.88 -1063.03
- Credit_History     1   23.2184 126.28  -947.06

Step:  AIC=-1071.81
Loan_Status ~ Gender + Married + Education + Self_Employed +
    ApplicantIncome + CoapplicantIncome + LoanAmount + Loan_Amount_Term +
    Credit_History + Property_Area

                    Df Sum of Sq    RSS     AIC
- Self_Employed      1    0.0006 103.06 -1073.81
- Gender             1    0.0054 103.06 -1073.78
- Loan_Amount_Term   1    0.0231 103.08 -1073.68
- ApplicantIncome    1    0.0460 103.10 -1073.54
- CoapplicantIncome  1    0.1830 103.24 -1072.73
<none>                          103.06 -1071.81
- Education          1    0.3674 103.42 -1071.63
- LoanAmount         1    0.5751 103.63 -1070.40
- Married            1    1.1397 104.20 -1067.06
- Property_Area      2    1.8263 104.88 -1065.03
- Credit_History     1   23.2224 126.28  -949.04

Step:  AIC=-1073.81
Loan_Status ~ Gender + Married + Education + ApplicantIncome +
    CoapplicantIncome + LoanAmount + Loan_Amount_Term + Credit_History +
    Property_Area

                    Df Sum of Sq    RSS     AIC
- Gender             1    0.0056 103.06 -1075.78
- Loan_Amount_Term   1    0.0233 103.08 -1075.67
- ApplicantIncome    1    0.0504 103.11 -1075.51
- CoapplicantIncome  1    0.1842 103.24 -1074.71
<none>                          103.06 -1073.81
- Education          1    0.3668 103.42 -1073.63
- LoanAmount         1    0.5772 103.64 -1072.38
- Married            1    1.1406 104.20 -1069.05
- Property_Area      2    1.8268 104.88 -1067.02
- Credit_History     1   23.2245 126.28  -951.03
```

```
Step:  AIC=-1075.78
Loan_Status ~ Married + Education + ApplicantIncome + CoapplicantIncome +
    LoanAmount + Loan_Amount_Term + Credit_History + Property_Area

                    Df Sum of Sq    RSS      AIC
- Loan_Amount_Term   1    0.0228 103.09 -1077.64
- ApplicantIncome    1    0.0485 103.11 -1077.49
- CoapplicantIncome  1    0.1791 103.24 -1076.71
<none>                           103.06 -1075.78
- Education          1    0.3766 103.44 -1075.54
- LoanAmount         1    0.5769 103.64 -1074.35
- Married            1    1.2031 104.27 -1070.65
- Property_Area      2    1.8727 104.94 -1068.72
- Credit_History     1   23.2695 126.33  -952.78

Step:  AIC=-1077.64
Loan_Status ~ Married + Education + ApplicantIncome + CoapplicantIncome +
    LoanAmount + Credit_History + Property_Area

                    Df Sum of Sq    RSS      AIC
- ApplicantIncome    1    0.0577 103.14 -1079.30
- CoapplicantIncome  1    0.1916 103.28 -1078.50
<none>                           103.09 -1077.64
- Education          1    0.3637 103.45 -1077.48
- LoanAmount         1    0.6186 103.70 -1075.97
- Married            1    1.2526 104.34 -1072.23
- Property_Area      2    1.8634 104.95 -1070.64
- Credit_History     1   23.2694 126.36  -954.67

Step:  AIC=-1079.3
Loan_Status ~ Married + Education + CoapplicantIncome + LoanAmount +
    Credit_History + Property_Area

                    Df Sum of Sq    RSS      AIC
- CoapplicantIncome  1    0.1356 103.28 -1080.49
<none>                           103.14 -1079.30
- Education          1    0.4168 103.56 -1078.82
- LoanAmount         1    0.6818 103.83 -1077.25
- Married            1    1.2512 104.39 -1073.89
- Property_Area      2    1.8341 104.98 -1072.48
- Credit_History     1   23.4931 126.64  -955.31
```

```
Step:  AIC=-1080.49
Loan_Status ~ Married + Education + LoanAmount + Credit_History +
    Property_Area

                 Df Sum of Sq    RSS      AIC
<none>                        103.28 -1080.49
- Education       1    0.4288 103.71 -1079.95
- LoanAmount      1    0.5830 103.86 -1079.03
- Property_Area   2    1.7989 105.08 -1073.89
- Married         1    1.4701 104.75 -1073.81
- Credit_History  1   23.3751 126.65  -957.22

Call:
lm(formula = Loan_Status ~ Married + Education + LoanAmount +
    Credit_History + Property_Area, data = Loan_model)

Coefficients:
   (Intercept)         Married       Education      LoanAmount  Credit_History  Property_Area1  Property_Area2
     0.2222284       0.1042197       0.0655541      -0.0005733       0.4689580       0.0417506       0.1287707

>
```

 # We start with all 11 predictors in the model.

 # For each backward step, the AIC column provides the model AIC resulting from the deletion

of the variable listed in that row.

 # As we can see when each variable is being removed the AIC value keeps on decreasing from

 # from -1069.82 to -1080.49.

 # Deleting any more variables would increase the AIC, so the process stops.

#Negative AIC indicates less information loss than a positive AIC and therefore a better model.

Finally the best model suggests that Loan Status has a strong relation to Credit History, Married, Property area, Loan Amount and Education.

So, I can conclude that my individual conclusion matched with the backward stepwise regression analysis.

CHECKING FOR CLASS IMBALANCE

prop.table(table(Loan_model$Loan_Status))

```
Console   Terminal ×
U:/R/R project/Final Project/
> prop.table(table(Loan_model$Loan_Status))

        0         1 
0.3127036 0.6872964 
> 
```

table(Loan_model$Loan_Status)

```
Console   Terminal ×
U:/R/R project/Final Project/
> table(Loan_model$Loan_Status)

  0   1 
192 422 
> 
```

#Class imbalance is a situation, mostly in classification model building; where the total number of

#positive class of a data set is extremely lower than the total number of the negative class.

#In the data set, we have 68.7% of the response variable as YES and 31.3% as NO.

#Hence, we can conclude that there is no class imbalance in this data set.

SPLITTING INTO TRAIN AND TEST DATA

set.seed(222)

split = sample(2,nrow(Loan_model),prob = c(0.75,0.25),replace = TRUE)

train_set = Loan_model[split == 1,]

test_set = Loan_model[split == 2,]

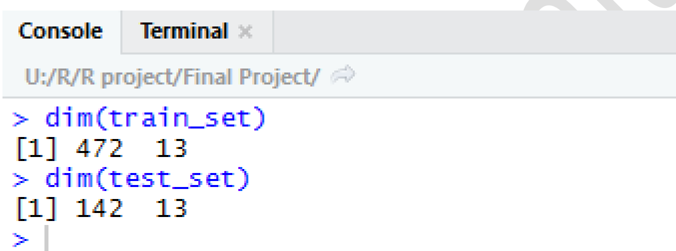
#It is the usual practice in Machine Learning field to divide the data set into train and test set.

#The model will be built on the train set and the performance of the model will be tested on the test.

#checking dimensions of train and test data sets

dim(train_set)

dim(test_set)



```
Console   Terminal ×
U:/R/R project/Final Project/
> dim(train_set)
[1] 472  13
> dim(test_set)
[1] 142  13
>
```

LOGISTIC REGRESSION

#Logistic regression uses sigmoid function to classify variables into classes

#and its basically applicable to classification problems

 # Fitting Logistic Regression to the Training set

logistics_classifier = glm(formula = Loan_Status ~ .,

family = binomial,

data = train_set[,-c(1)])

summary(logistics_classifier)

Console Terminal

U:/R/R project/Final Project/

```
> logistics_classifier = glm(formula = Loan_Status ~ .,
+                              family = binomial,
+                              data = train_set[,-c(1)])
> summary(logistics_classifier)

Call:
glm(formula = Loan_Status ~ ., family = binomial, data = train_set[,
    -c(1)])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3670  -0.8250   0.5534   0.7176   1.9668

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)       -1.305e+00  8.060e-01  -1.619  0.10534
Gender            -2.591e-01  2.991e-01  -0.866  0.38634
Married            7.798e-01  2.688e-01   2.901  0.00372 **
Dependents        -6.123e-03  1.228e-01  -0.050  0.96024
Education          3.530e-01  2.743e-01   1.287  0.19820
Self_Employed      4.106e-01  3.692e-01   1.112  0.26610
ApplicantIncome    1.823e-05  7.133e-05   0.256  0.79832
CoapplicantIncome  1.079e-04  8.721e-05   1.237  0.21620
LoanAmount        -5.064e-03  3.053e-03  -1.659  0.09709 .
Loan_Amount_Term   6.152e-06  1.902e-03   0.003  0.99742
Credit_History     2.226e+00  2.581e-01   8.624  < 2e-16 ***
Property_Area1     2.711e-01  2.791e-01   0.971  0.33138
Property_Area2     6.399e-01  2.825e-01   2.265  0.02349 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 591.70  on 471  degrees of freedom
Residual deviance: 484.35  on 459  degrees of freedom
AIC: 510.35

Number of Fisher Scoring iterations: 4
```

#Based on the output of the Logistic regression,only 4 variables are significant while other are insignificant.

#Credit_History is an important factor in deciding whether a client will default or not

#and this was clearly in tune with the outcome of the model.

#Whether the customer is married or not is also a significant factor, as far as this data set is concerned.

#Property_Area2 and Loan Amount are also significant factors after the above mentioned two attributes.

PREDICTION USING LOGISTICS REGRESSOR

 # Predicting the Test set results

prob_pred = predict(logistics_classifier, type = 'response', newdata = test_set)

y_pred = ifelse(prob_pred > 0.5, 1, 0)

dim(output)

```
Console   Terminal
U:/R/R project/Final Project/
> dim(output)
[1] 142  14
>
```

output <- cbind(test_set, My_pred)

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status	My_pred
1	LP001002	1	0	0	1	0	5849.00	0.000	146.4122	360	1	1	1	1
5	LP001008	1	0	0	1	0	6000.00	0.000	141.0000	360	1	1	1	1
6	LP001011	1	1	2	1	1	5417.00	4196.000	261.5500	360	1	1	1	1
15	LP001030	1	1	2	1	0	1299.00	1086.000	17.0000	120	1	1	1	1
21	LP001043	1	1	0	0	0	7660.00	0.000	104.0000	360	0	1	0	0
26	LP001066	1	1	0	1	1	9560.00	0.000	191.0000	360	1	2	1	1
30	LP001087	0	0	2	1	0	3750.00	2083.000	120.0000	360	1	2	1	1
36	LP001106	1	1	0	1	0	2275.00	2067.000	146.4122	360	1	1	1	1
40	LP001116	1	0	0	0	0	3748.00	1668.000	110.0000	360	1	2	1	1
41	LP001119	1	0	0	1	0	3600.00	0.000	80.0000	360	1	1	0	1
42	LP001120	1	0	0	1	0	1800.00	1213.000	47.0000	360	1	1	1	1
43	LP001123	1	1	0	1	0	2400.00	0.000	75.0000	360	0	1	1	0
47	LP001138	1	1	1	1	0	5649.00	0.000	44.0000	360	1	1	1	1
50	LP001151	0	0	0	1	0	4000.00	2275.000	144.0000	360	1	2	1	1
54	LP001179	1	1	2	1	0	4616.00	0.000	134.0000	360	1	1	0	1
59	LP001198	1	1	1	1	0	8080.00	2250.000	180.0000	360	1	1	1	1

CONFUSION MATRIX

#estimating the performance of the model

cm = table(ActualValue=test_set$Loan_Status, PredictedValue=prob_pred > 0.5)

cm

```
Console   Terminal
U:/R/R project/Final Project/
> cm = table(ActualValue=test_set$Loan_Status, PredictedValue=prob_pred > 0.5)
> cm
            PredictedValue
ActualValue FALSE TRUE
          0    18   23
          1     9   92
>
```

#We can check by building a confusion matrix to display the success rate of

#our model's predictions on the testing data we created earlier.

#The table function builds the confusion matrix. Going diagonally, (18, 92) represent the number of correct predictions.

#Conversely, the going up diagonally, (9, 23) represent the number of incorrect predictions.

ESTIMATING THE PERCENTAGE OF PERFORMANCE

sum(diag(cm))/sum(cm)

```
Console   Terminal ×
U:/R/R project/Final Project/
> sum(diag(cm))/sum(cm)
[1] 0.7746479
>
```

#Logistics Regression was able to give us an accuracy of 77.46%,

#which means that we can expect our model to classify correct about 8 observations in every 10.