# LOAN STATUS PREDICTION

cris

7/29/2024

STEP 1: READIND AND UNDERSTANDING THE DATA

```
# Clear the memory and check the path for the data
rm(list=ls())
getwd()

## [1] "C:/Users/hp/Desktop/Projects/loan prediction using machine learning"

# Change the path to the desired one
setwd("C:\\Users\\hp\\Desktop\\Projects\\loan prediction using machine
learning")
getwd()

## [1] "C:/Users/hp/Desktop/Projects/loan prediction using machine learning"

# Read the data and print the first 6 variables
data = read.csv("loan-train.csv")
head(data)

##    Loan_ID Gender Married Dependents    Education Self_Employed
ApplicantIncome
## 1 LP001002   Male      No          0     Graduate            No
5849
## 2 LP001003   Male     Yes          1     Graduate            No
4583
## 3 LP001005   Male     Yes          0     Graduate           Yes
3000
## 4 LP001006   Male     Yes          0 Not Graduate            No
2583
## 5 LP001008   Male      No          0     Graduate            No
6000
## 6 LP001011   Male     Yes          2     Graduate           Yes
5417
##   CoapplicantIncome LoanAmount Loan_Amount_Term Credit_History
Property_Area
## 1                 0         NA              360              1
Urban
## 2              1508        128              360              1
Rural
## 3                 0         66              360              1
Urban
## 4              2358        120              360              1
Urban
```

```
## 5                 0       141              360              1
Urban
## 6              4196       267              360              1
Urban
##   Loan_Status
## 1           Y
## 2           N
## 3           Y
## 4           Y
## 5           Y
## 6           Y
```

```r
# Import the necessary libraries
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

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

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

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

```r
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.3
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
## Warning: package 'tibble' was built under R version 4.1.3
```

```
## Warning: package 'tidyr' was built under R version 4.1.3
```

```
## Warning: package 'readr' was built under R version 4.1.3
```

```
## Warning: package 'purrr' was built under R version 4.1.3
```

```
## Warning: package 'stringr' was built under R version 4.1.3
```

```
## Warning: package 'forcats' was built under R version 4.1.3
```

```
## Warning: package 'lubridate' was built under R version 4.1.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse
2.0.0 --
## v forcats   1.0.0     v readr     2.1.4
## v ggplot2   3.4.2     v stringr   1.5.0
```

```
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0
```

```
## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

```r
library(ggplot2)
library(stargazer)
```

```
## Warning: package 'stargazer' was built under R version 4.1.2
```

```
##
## Please cite as:
##
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary
Statistics Tables.
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```r
# view the structure of our dataset
str(data)
```

```
## 'data.frame':    614 obs. of  13 variables:
##  $ Loan_ID          : chr  "LP001002" "LP001003" "LP001005" "LP001006" ...
##  $ Gender           : chr  "Male" "Male" "Male" "Male" ...
##  $ Married          : chr  "No" "Yes" "Yes" "Yes" ...
##  $ Dependents       : chr  "0" "1" "0" "0" ...
##  $ Education        : chr  "Graduate" "Graduate" "Graduate" "Not Graduate"
...
##  $ Self_Employed    : chr  "No" "No" "Yes" "No" ...
##  $ 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    : chr  "Urban" "Rural" "Urban" "Urban" ...
##  $ Loan_Status      : chr  "Y" "N" "Y" "Y" ...
```

```r
# summarize the dataset in a table format
stargazer(data, type = "text")
```

```
##
## =========================================================
## Statistic          N    Mean    St. Dev.   Min     Max
## ---------------------------------------------------------
## ApplicantIncome   614 5,403.459 6,109.042  150    81,000
## CoapplicantIncome 614 1,621.246 2,926.248 0.000 41,667.000
## LoanAmount        592  146.412   85.587     9      700
```

```
## Loan_Amount_Term  600  342.000   65.120    12      480
## Credit_History    564   0.842     0.365     0       1
## ----------------------------------------------------------
```

The first row 'N' shows that there are missing values int the dataset However, ApplicantIncome and CoapplicantIncome seems to be right skewed due to their large standard deviation. Credit histry has only two values, 0 and 1 hence its not an integre but rather a factor with two level , ie, the applicant has either taken a loan before or not. Loan_Amount_Term appears to be symetrical.

```
summary(data)
```

```
##    Loan_ID             Gender             Married            Dependents
##   Length:614         Length:614         Length:614         Length:614
##   Class :character   Class :character   Class :character   Class :character
##   Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   Education          Self_Employed      ApplicantIncome CoapplicantIncome
##   Length:614         Length:614         Min.   :  150   Min.   :    0
##   Class :character   Class :character   1st Qu.: 2878   1st Qu.:    0
##   Mode  :character   Mode  :character   Median : 3812   Median : 1188
##                                         Mean   : 5403   Mean   : 1621
##                                         3rd Qu.: 5795   3rd Qu.: 2297
##                                         Max.   :81000   Max.   :41667
##
##    LoanAmount     Loan_Amount_Term Credit_History   Property_Area
##   Min.   :  9.0   Min.   : 12      Min.   :0.0000   Length:614
##   1st Qu.:100.0   1st Qu.:360      1st Qu.:1.0000   Class :character
##   Median :128.0   Median :360      Median :1.0000   Mode  :character
##   Mean   :146.4   Mean   :342      Mean   :0.8422
##   3rd Qu.:168.0   3rd Qu.:360      3rd Qu.:1.0000
##   Max.   :700.0   Max.   :480      Max.   :1.0000
##   NA's   :22      NA's   :14       NA's   :50
##   Loan_Status
##   Length:614
##   Class :character
##   Mode  :character
##
##
##
##
```

STEP 2: DATA CLEANING. First we check for missing values.

```
# Lets check for missing values
any(is.na(data))
```

```
## [1] TRUE
```

We have missing values. lets check how many missing values do we have

```
sum(is.na(data))

## [1] 86

# now, we check if we have empty entries and convert them to NA
colSums(is.na(data)|data == "")

##            Loan_ID            Gender           Married        Dependents
##                  0                13                 3                15
##          Education     Self_Employed   ApplicantIncome CoapplicantIncome
##                  0                32                 0                 0
##         LoanAmount  Loan_Amount_Term    Credit_History     Property_Area
##                 22                14                50                 0
##        Loan_Status
##                  0

# gender, married, dependents,Self_Employed, LoanAmount, Loan_Amount_Term and
Credit_History have emty entries. we then convert them to NA
data[data == ""] = NA
sum(is.na(data))

## [1] 149
```

We then handle the missing values

```
data2 = na.omit(data)
sum(is.na(data2))

## [1] 0
```

We have no missing valus, nesxt we deal with datatypes

```
# first, we convert character variable to factor form
data3 =  as.data.frame(unclass(data2), stringsAsFactors = TRUE)
data3$Credit_History = as.factor(data3$Credit_History)
str(data3)

## 'data.frame':    480 obs. of  13 variables:
##  $ Loan_ID          : Factor w/ 480 levels "LP001003","LP001005",..: 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": 2 2 2 1 2 2 2 2 2 2
...
##  $ Dependents       : Factor w/ 4 levels "0","1","2","3+": 2 1 1 1 3 1 4 3
2 3 ...
##  $ Education        : Factor w/ 2 levels "Graduate","Not Graduate": 1 1 2
1 1 2 1 1 1 1 ...
##  $ Self_Employed    : Factor w/ 2 levels "No","Yes": 1 2 1 1 2 1 1 1 1 1
...
```

```
## $ ApplicantIncome  : int  4583 3000 2583 6000 5417 2333 3036 4006 12841
3200 ...
## $ CoapplicantIncome: num  1508 0 2358 0 4196 ...
## $ LoanAmount       : int  128 66 120 141 267 95 158 168 349 70 ...
## $ Loan_Amount_Term : int  360 360 360 360 360 360 360 360 360 360 ...
## $ Credit_History   : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 1 2 2 2 ...
## $ Property_Area    : Factor w/ 3 levels "Rural","Semiurban",..: 1 3 3 3 3
3 2 3 2 3 ...
## $ Loan_Status      : Factor w/ 2 levels "N","Y": 1 2 2 2 2 2 1 2 1 2 ...
```

Now lets check for duplicates

```
duplicates = duplicated(data3)|duplicated(data3, fromlast = TRUE)
any(duplicates)

## [1] FALSE
```

The data is now clean for analysis

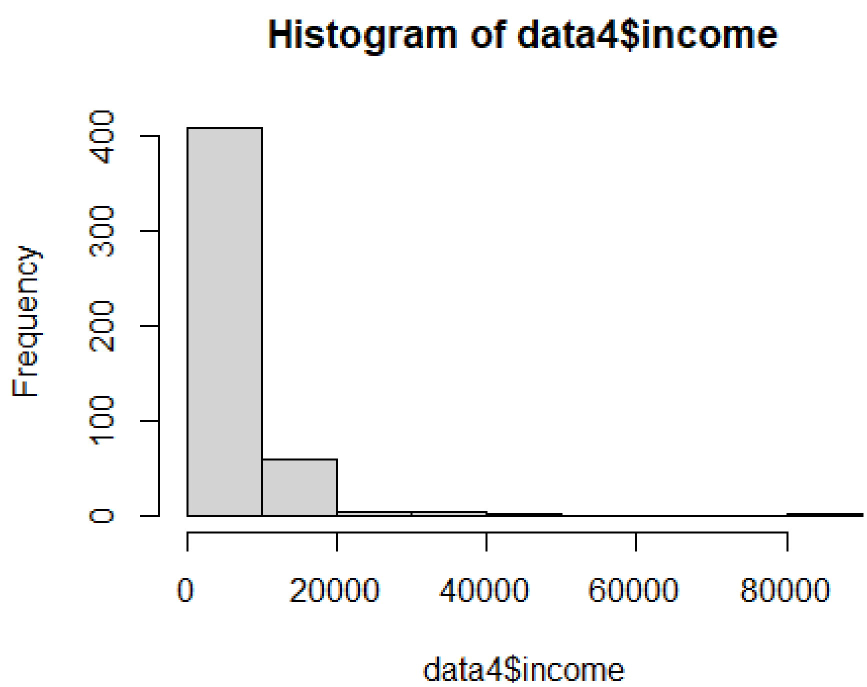
STEP 3: DATA VISUALIZATION (EDA)

```
# Combine columns for income
data4 = data3 %>% mutate(income = data3$ApplicantIncome +
data3$CoapplicantIncome)
head(data4)

##    Loan_ID Gender Married Dependents    Education Self_Employed
ApplicantIncome
## 1 LP001003   Male     Yes          1     Graduate            No
4583
## 2 LP001005   Male     Yes          0     Graduate           Yes
3000
## 3 LP001006   Male     Yes          0 Not Graduate            No
2583
## 4 LP001008   Male      No          0     Graduate            No
6000
## 5 LP001011   Male     Yes          2     Graduate           Yes
5417
## 6 LP001013   Male     Yes          0 Not Graduate            No
2333
##   CoapplicantIncome LoanAmount Loan_Amount_Term Credit_History
Property_Area
## 1              1508        128              360              1
Rural
## 2                 0         66              360              1
Urban
## 3              2358        120              360              1
Urban
## 4                 0        141              360              1
Urban
## 5              4196        267              360              1
```

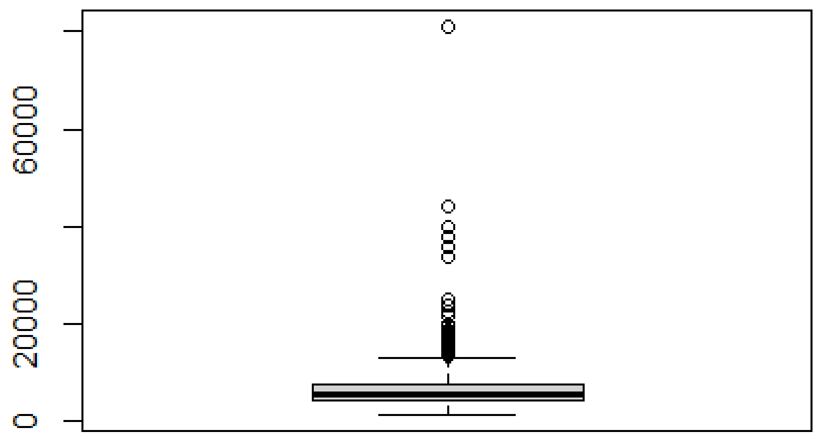
```
Urban
## 6             1516         95               360              1
Urban
##   Loan_Status income
## 1           N   6091
## 2           Y   3000
## 3           Y   4941
## 4           Y   6000
## 5           Y   9613
## 6           Y   3849

# Now lets check the distribution for income
hist(data4$income)
```




We can see that income is not normally distributed, ie, it is right skewed. Most of the data tend to lie below 20000

```
# Visualizing the boxplot
boxplot(data4$income)
```




The variable income has also some outliers. lets try to handle the case for outliers and normality by using logarithmic transformation.

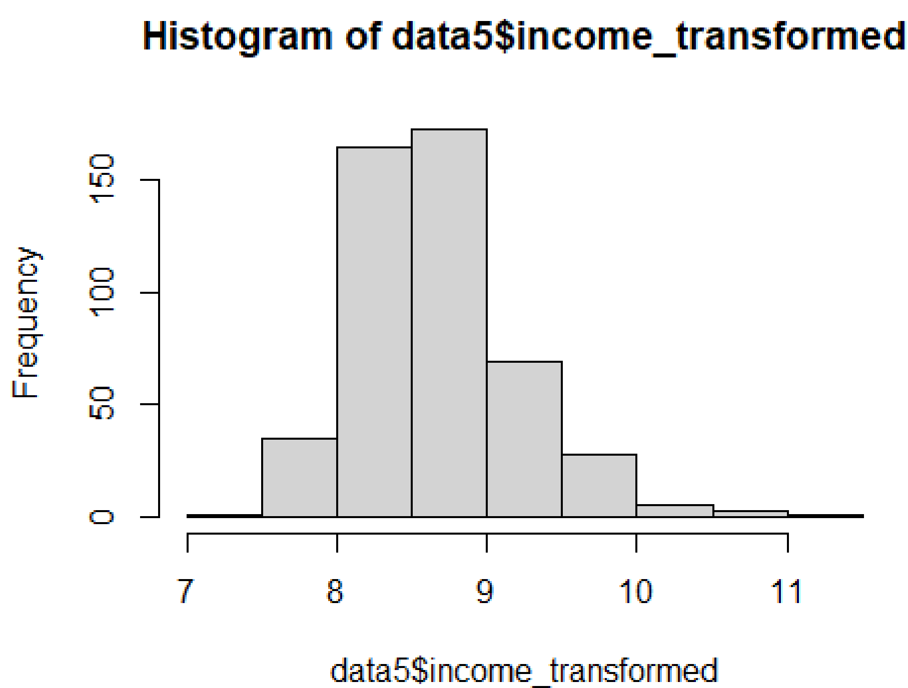
```
data5 = data4 %>% mutate(income_transformed = log(income))
head(data5)

##    Loan_ID Gender Married Dependents    Education Self_Employed
ApplicantIncome
## 1 LP001003   Male     Yes          1     Graduate            No
4583
## 2 LP001005   Male     Yes          0     Graduate           Yes
3000
## 3 LP001006   Male     Yes          0 Not Graduate            No
2583
## 4 LP001008   Male      No          0     Graduate            No
6000
## 5 LP001011   Male     Yes          2     Graduate           Yes
5417
## 6 LP001013   Male     Yes          0 Not Graduate            No
2333
##   CoapplicantIncome LoanAmount Loan_Amount_Term Credit_History
Property_Area
## 1              1508        128              360              1
Rural
## 2                 0         66              360              1
Urban
## 3              2358        120              360              1
```

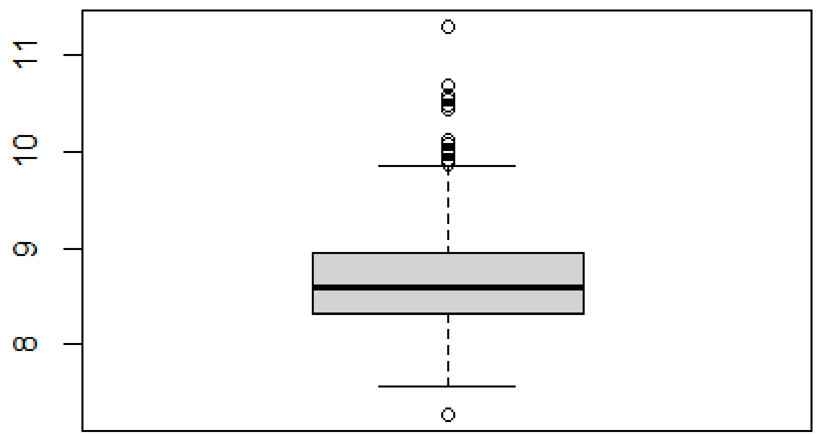
```
Urban
## 4                    0        141              360              1
Urban
## 5                 4196        267              360              1
Urban
## 6                 1516         95              360              1
Urban
##   Loan_Status income income_transformed
## 1           N   6091           8.714568
## 2           Y   3000           8.006368
## 3           Y   4941           8.505323
## 4           Y   6000           8.699515
## 5           Y   9613           9.170872
## 6           Y   3849           8.255569
```

check the privious distribution

```
hist(data5$income_transformed)
```

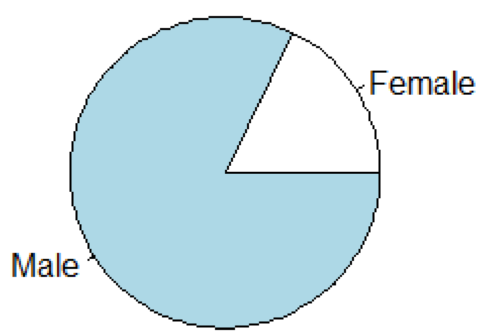



```
boxplot(data5$income_transformed)
```

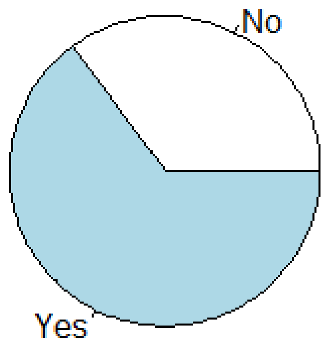



Income now looks to be normal with a reduced number of outliers lets check some other sitributions

```
pie(table(data5$Gender))
```



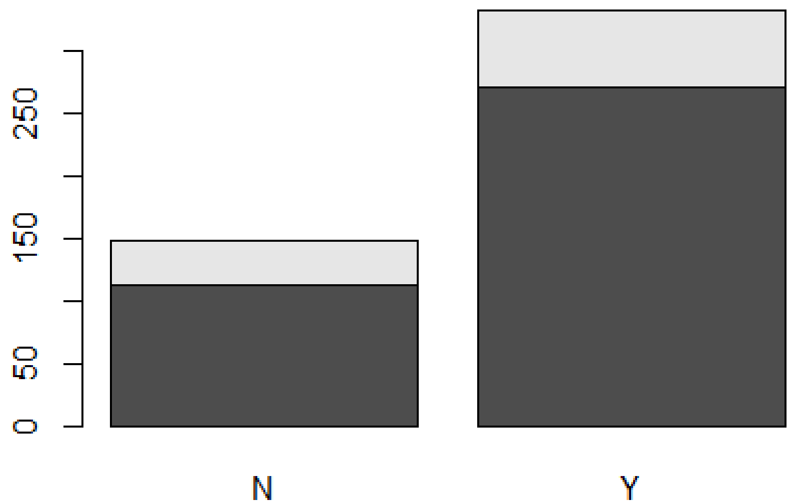
```r
pie(table(data5$Married))
```




```r
table(data5$Education, data5$Loan_Status)

##
##                  N   Y
##   Graduate     112 271
##   Not Graduate  36  61

barplot(table(data5$Education, data5$Loan_Status))
```

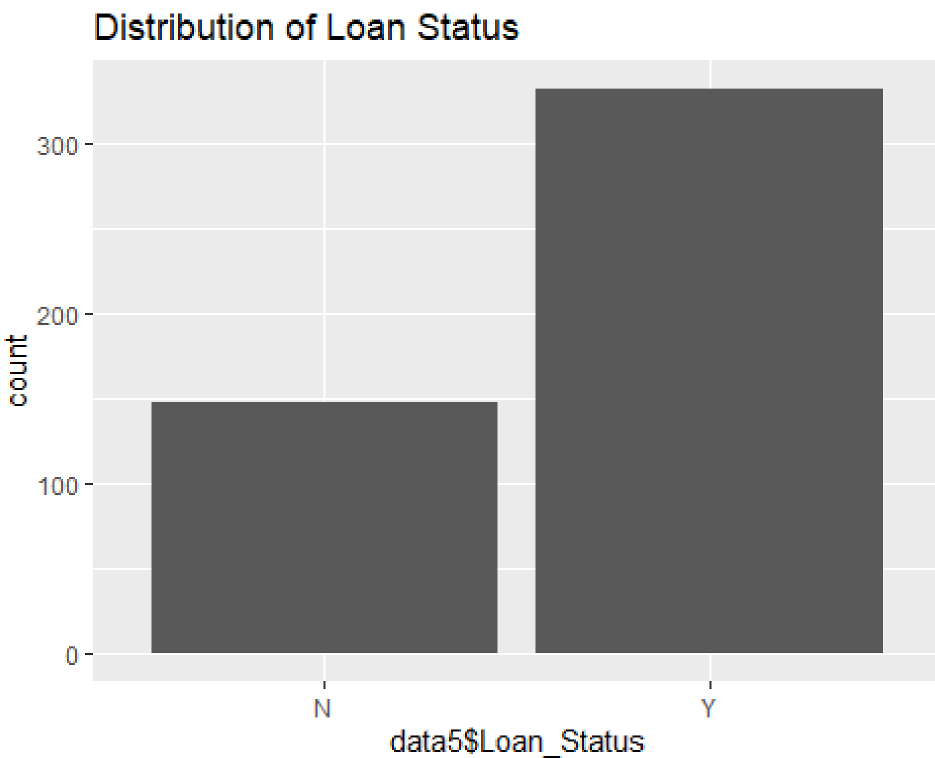



```r
table(data5$Dependents, data5$Loan_Status)

##
##        N   Y
##   0   87 187
##   1   28  52
##   2   20  65
##   3+  13  28

# Distribution of loan status
ggplot(data5, aes(x= data5$Loan_Status))+geom_bar()+ggtitle("Distribution of
Loan Status")

## Warning: Use of `data5$Loan_Status` is discouraged.
## i Use `Loan_Status` instead.
```




STEP 4: DATA PREPROCESSING

```
#we first driop the loan id column income, ApplicantIncome and
CoapplicantIncome since we no longer need them
data6 = data5[, !(names(data5) %in%
c("Loan_ID","income","ApplicantIncome","CoapplicantIncome"))]
head(data6)

##   Gender Married Dependents    Education Self_Employed LoanAmount
## 1   Male     Yes          1     Graduate            No        128
## 2   Male     Yes          0     Graduate           Yes         66
## 3   Male     Yes          0 Not Graduate            No        120
## 4   Male      No          0     Graduate            No        141
## 5   Male     Yes          2     Graduate           Yes        267
## 6   Male     Yes          0 Not Graduate            No         95
##   Loan_Amount_Term Credit_History Property_Area Loan_Status
income_transformed
## 1              360              1         Rural           N
8.714568
## 2              360              1         Urban           Y
8.006368
## 3              360              1         Urban           Y
8.505323
## 4              360              1         Urban           Y
8.699515
## 5              360              1         Urban           Y
9.170872
```

```
## 6                360               1          Urban            Y
8.255569
```

```r
# checking the number of observations
dim(data6)
```

```
## [1] 480  11
```

```r
# dividing data into training (80%) and testing (20%)
train_split = data6[1:384,]
dim(train_split)
```

```
## [1] 384  11
```

```r
test_split = data6[385:480,]
dim(test_split)
```

```
## [1] 96 11
```

```r
# Apply feature scaling for numeric variables
scale_features = function(x){
  num_col = sapply(x, is.numeric)
  x[num_col] = scale(x[num_col])
  return(x)
}

scaled_train = scale_features(train_split)
scaled_test = scale_features(test_split)
```

STEP 5: MODEL FITTING We will comapare different models and select the best one from 1. logistic regression 2. naive bayes 4. random forest

```r
# first, we will use logistic regression
attach(scaled_train)
model = glm(Loan_Status ~ ., family = binomial(link = 'logit'), data =
scaled_train)
detach(scaled_train)

# printing the summary statistics for our logistc model
summary(model)
```

```
##
## Call:
## glm(formula = Loan_Status ~ ., family = binomial(link = "logit"),
##     data = scaled_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2654  -0.4230   0.4547   0.6942   2.3520
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)           -3.05921    0.59867  -5.110 3.22e-07 ***
## GenderMale             0.44288    0.37233   1.189  0.23425
## MarriedYes             0.37696    0.32423   1.163  0.24498
## Dependents1           -0.66458    0.37989  -1.749  0.08022 .
## Dependents2            0.37682    0.43653   0.863  0.38802
## Dependents3+           0.26450    0.54571   0.485  0.62790
## EducationNot Graduate  -0.38808    0.35871  -1.082  0.27931
## Self_EmployedYes      -0.44277    0.41806  -1.059  0.28955
## LoanAmount            -0.57555    0.20238  -2.844  0.00446 **
## Loan_Amount_Term      -0.28462    0.17170  -1.658  0.09738 .
## Credit_History1        3.62633    0.48590   7.463 8.45e-14 ***
## Property_AreaSemiurban  1.08703    0.34975   3.108  0.00188 **
## Property_AreaUrban     -0.00508    0.33902  -0.015  0.98805
## income_transformed     0.33122    0.21862   1.515  0.12975
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 478.56  on 383  degrees of freedom
## Residual deviance: 342.87  on 370  degrees of freedom
## AIC: 370.87
##
## Number of Fisher Scoring iterations: 5
```

Lets explain the results; 1. We can see that, only the Intercept, LoanAmount, Credit_History1 and Property_AreaSemiurban are significant in predicting the probality of a loan to be approved due to their loww p-value (<0.05). The main significant variable is Credit_History1 showing that having a credit history strongly increases the likelihood of loan approval.

2.The intercept (-5.710799) is the baseline log-odds of loan approval when all the predictors are set to 0.

3. At the top, we have a summary of the devince residual which is a measure of goodness of fit for the model. Since they are close to being centered on, they look symmetrical

4. The coeeficient for the GenderMale (0,442876) shows that, being male increases the log-odds of loan approval compare to the baseline category (female)

5. For education, not being a graduate reduces the log-odd of loan approval by 0.388078.

Based on the above explanations, the other variables applies the same explanations compared to the respective baseline categories.

The null deviance is the deviance of the model with only the intercept The residual deviance is the model with the predictors included The AIC (Akaike Information Criterion) is the is a measure of the model quality.

```r
# performing a chi-square tets for the model
anova(model, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Loan_Status
##
## Terms added sequentially (first to last)
##
##
##                    Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                 383     478.56
## Gender              1    2.810       382     475.75 0.0936674 .
## Married             1    1.644       381     474.10 0.1997343
## Dependents          3    6.221       378     467.88 0.1013352
## Education           1    2.266       377     465.62 0.1322418
## Self_Employed       1    0.773       376     464.84 0.3792810
## LoanAmount          1    9.398       375     455.45 0.0021720 **
## Loan_Amount_Term    1    1.135       374     454.31 0.2867354
## Credit_History      1   94.589       373     359.72 < 2.2e-16 ***
## Property_Area       2   14.444       371     345.28 0.0007304 ***
## income_transformed  1    2.414       370     342.87 0.1202912
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# now lets apply the fitted model to the test dataset
fitted.results = predict(model, scaled_train, type = 'response')
fitted.results = ifelse(fitted.results > 0.5,1,0)
misClasificError = mean(fitted.results != scaled_train$Loan_Status)
head(fitted.results)
```

```
## 1 2 3 4 5 6
## 1 1 1 1 1 1
```

STEP 5: This the last step it involves ploting the ROC curve and calculating the AUC.

TO BE CONTINUED…..