DANA 4820 - group 6 Project

Importing the Libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(mice) #if needed install this pacakge
```

```
##
## Attaching package: 'mice'
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
library(tidyverse)
library(fastDummies)
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(leaps)
library(here)
```

```
## here() starts at /Users/arath/Documents/langara/DANA-4820
```

```
library(skimr)
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(lubridate)
```

```
## Loading required package: timechange
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(LaplacesDemon)
```

```
##
## Attaching package: 'LaplacesDemon'
##
## The following objects are masked from 'package:lubridate':
##
##     dst, interval
##
## The following object is masked from 'package:purrr':
##
##     partial
```

```
library(WVPlots)
```

```
## Loading required package: wrapr
##
## Attaching package: 'wrapr'
##
## The following object is masked from 'package:dplyr':
##
##     coalesce
##
## The following objects are masked from 'package:tidyr':
##
##     pack, unpack
##
## The following object is masked from 'package:tibble':
##
##     view
```

```
library(praznik)
library(standardize)
```

```
##
##  ************************************************************
##           Loading standardize package version 0.2.2
##      Call standardize.news() to see new features/changes
##  ************************************************************
```

```
library(clusterSim)
```

```
## Loading required package: cluster
```

```
library(dplyr)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(caTools)
library(ggcorrplot)
library(Metrics)
```

```
##
## Attaching package: 'Metrics'
##
## The following objects are masked from 'package:caret':
##
##     precision, recall
```

```
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:wrapr':
##
##     bc
##
## The following object is masked from 'package:LaplacesDemon':
##
##     logit
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```
library(olsrr)
```

```
##
## Attaching package: 'olsrr'
##
## The following object is masked from 'package:MASS':
##
##     cement
##
## The following object is masked from 'package:datasets':
##
##     rivers
```

```
library(PerformanceAnalytics)
```

```
## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
##
## Attaching package: 'xts'
##
## The following objects are masked from 'package:dplyr':
##
##     first, last
##
##
## Attaching package: 'PerformanceAnalytics'
##
## The following object is masked from 'package:graphics':
##
##     legend
```

```
library(sjPlot)
library(sjmisc)
```

```
##
## Attaching package: 'sjmisc'
##
## The following objects are masked from 'package:janitor':
##
##     remove_empty_cols, remove_empty_rows
##
## The following object is masked from 'package:skimr':
##
##     to_long
##
## The following object is masked from 'package:purrr':
##
##     is_empty
```

```
##
## The following object is masked from 'package:tidyr':
##
##     replace_na
##
## The following object is masked from 'package:tibble':
##
##     add_case
```

```
library(ggplot2)
```

#1. Data overview #we are usisng the following data for our analysis https://www.kaggle.com/datasets/yasserh/loan-default-dataset Banks earn a major revenue from lending loans. But it is often associated with risk. The borrower's may default on the loan. To mitigate this issue, the banks have decided to use Machine Learning to overcome this issue. They have collected past data on the loan borrowers & would like you to develop a strong ML Model to classify if any new borrower is likely to default or not.

The dataset is enormous & consists of multiple deteministic factors like borrowe's income, gender, loan pupose etc. The dataset is subject to strong multicollinearity & empty values.

```
df <- read.csv("Loan_default.csv")
str(df)
```

```
## 'data.frame':    148670 obs. of  34 variables:
##  $ ID                       : int  24890 24891 24892 24893 24894 24895 24896 24897 24898 24899 ...
##  $ year                     : int  2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 ...
##  $ loan_limit               : chr  "cf" "cf" "cf" "cf" ...
##  $ Gender                   : chr  "Sex Not Available" "Male" "Male" "Male" ...
##  $ approv_in_adv            : chr  "nopre" "nopre" "pre" "nopre" ...
##  $ loan_type                : chr  "type1" "type2" "type1" "type1" ...
##  $ loan_purpose             : chr  "p1" "p1" "p1" "p4" ...
##  $ Credit_Worthiness        : chr  "l1" "l1" "l1" "l1" ...
##  $ open_credit              : chr  "nopc" "nopc" "nopc" "nopc" ...
##  $ business_or_commercial   : chr  "nob/c" "b/c" "nob/c" "nob/c" ...
##  $ loan_amount              : int  116500 206500 406500 456500 696500 706500 346500 266500 376500 43
##  $ rate_of_interest         : num  NA NA 4.56 4.25 4 ...
##  $ Interest_rate_spread     : num  NA NA 0.2 0.681 0.304 ...
##  $ Upfront_charges          : num  NA NA 595 NA 0 ...
##  $ term                     : int  360 360 360 360 360 360 360 360 360 360 ...
##  $ Neg_ammortization        : chr  "not_neg" "not_neg" "neg_amm" "not_neg" ...
##  $ interest_only            : chr  "not_int" "not_int" "not_int" "not_int" ...
##  $ lump_sum_payment         : chr  "not_lpsm" "lpsm" "not_lpsm" "not_lpsm" ...
##  $ property_value           : int  118000 NA 508000 658000 758000 1008000 438000 308000 478000 68800
##  $ construction_type        : chr  "sb" "sb" "sb" "sb" ...
##  $ occupancy_type           : chr  "pr" "pr" "pr" "pr" ...
##  $ Secured_by               : chr  "home" "home" "home" "home" ...
##  $ total_units              : chr  "1U" "1U" "1U" "1U" ...
##  $ income                   : int  1740 4980 9480 11880 10440 10080 5040 3780 5580 6720 ...
##  $ credit_type              : chr  "EXP" "EQUI" "EXP" "EXP" ...
##  $ Credit_Score             : int  758 552 834 587 602 864 860 863 580 788 ...
##  $ co.applicant_credit_type : chr  "CIB" "EXP" "CIB" "CIB" ...
##  $ age                      : chr  "25-34" "55-64" "35-44" "45-54" ...
##  $ submission_of_application: chr  "to_inst" "to_inst" "to_inst" "not_inst" ...
##  $ LTV                      : num  98.7 NA 80 69.4 91.9 ...
##  $ Region                   : chr  "south" "North" "south" "North" ...
##  $ Security_Type            : chr  "direct" "direct" "direct" "direct" ...
```

```
##  $ Status                   : int  1 1 0 0 0 0 0 0 0 0 ...
##  $ dtir1                    : int  45 NA 46 42 39 40 44 42 44 30 ...
```

#Based on the data provider here are the description of the columns

ID = Customer ID of Applicant year = Year of Application loan limit = maximum avaliable amount of the loan allowed to be taken Gender = sex type approv_in_adv = Is loan pre-approved or not loan_type = Type of loan loan_purpose = the reason you want to borrow money Credit_Worthiness = is how a lender determines that you will default on your debt obligations, or how worthy you are to receive new credit. open_credit = is a pre-approved loan between a lender and a borrower. It allows the borrower to make repeated withdrawals up to a certain limit. business_or_commercial = Usage type of the loan amount loan_amount = The exact loan amount rate_of_interest = is the amount a lender charges a borrower and is a percentage of the principal—the amount loaned. Interest_rate_spread = the difference between the interest rate a financial institution pays to depositors and the interest rate it receives from loans Upfront_charges = Fee paid to a lender by a borrower as consideration for making a new loan term = the loan's repayment period Neg_ammortization = refers to a situation when a loan borrower makes a payment less than the standard installment set by the bank. interest_only = amount of interest only without principles lump_sum_payment = is an amount of money that is paid in one single payment rather than in installments. property_value = the present worth of future benefits arising from the ownership of the property construction_type = Collateral construction type occupancy_type = classifications refer to categorizing structures based on their usage Secured_by = Type of Collatoral total_units = number of unites income = refers to the amount of money, property, and other transfers of value received over a set period of time credit_type = type of credit co-applicant_credit_type = is an additional person involved in the loan application process. Both applicant and co-applicant apply and sign for the loan age = applicant's age submission_of_application = Ensure the application is complete or not LTV = life-time value (LTV) is a prognostication of the net profit Region = applicant's place Security_Type = Type of Collatoral status = Loan status (Approved/Declined) dtir1 = debt-to-income ratio

#Based on our investigation 7 Factors Lenders Look at When Considering Loan Application 1. credit score. ... 2. income and employment history. ... 3. debt-to-income ratio. ... 4. Value of your collateral. ... 5. Size of down payment. ... 6. Liquid assets. ... 7. Loan term.

We should use the following variables as our explanatory variable 'loan_type', 'loan_amount', 'rate_of_interest', 'term', 'property_value', 'income', 'Credit_Score', 'age', 'dtir1'

and Status as our predictor

#lets create smaller df with the variables of our intrest

```
df_reduced <- df[,c('Status', 'loan_type', 'loan_amount', 'rate_of_interest', 'term','property_value',
summary(df_reduced)
```

```
##      Status        loan_type          loan_amount      rate_of_interest
##  Min.   :0.0000   Length:148670      Min.   :  16500   Min.   :0.00
##  1st Qu.:0.0000   Class :character   1st Qu.: 196500   1st Qu.:3.62
##  Median :0.0000   Mode  :character   Median : 296500   Median :3.99
##  Mean   :0.2464                      Mean   : 331118   Mean   :4.05
##  3rd Qu.:0.0000                      3rd Qu.: 436500   3rd Qu.:4.38
##  Max.   :1.0000                      Max.   :3576500   Max.   :8.00
##                                                        NA's   :36439
##       term       property_value         income        Credit_Score
##  Min.   : 96.0   Min.   :    8000   Min.   :     0   Min.   :500.0
##  1st Qu.:360.0   1st Qu.:  268000   1st Qu.:  3720   1st Qu.:599.0
##  Median :360.0   Median :  418000   Median :  5760   Median :699.0
##  Mean   :335.1   Mean   :  497893   Mean   :  6957   Mean   :699.8
##  3rd Qu.:360.0   3rd Qu.:  628000   3rd Qu.:  8520   3rd Qu.:800.0
##  Max.   :360.0   Max.   :16508000   Max.   :578580   Max.   :900.0
##  NA's   :41      NA's   :15098      NA's   :9150
```

```
##      age                dtir1
##  Length:148670      Min.   : 5.00
##  Class :character   1st Qu.:31.00
##  Mode  :character   Median :39.00
##                     Mean   :37.73
##                     3rd Qu.:45.00
##                     Max.   :61.00
##                     NA's   :24121
```

#2. Data Cleanup #lets find missing values for the variables of our interest

```
missing <-sapply(df_reduced, function(y) sum(length(which(is.na(y)))))
missing <- data.frame(missing)
missing
```

```
##                   missing
## Status                  0
## loan_type               0
## loan_amount             0
## rate_of_interest    36439
## term                   41
## property_value      15098
## income               9150
## Credit_Score            0
## age                     0
## dtir1               24121
```

#lets impute the misisng values based on other variables impute property_value based on loan_amount and loan_type impute term based on loan_amount and loan_type impute rate_of_interest based on loan_amount, loan_type, Term impute income based on Age impute dtir1 based property_value, loan_amount and loan_type

Since we have dependent variables missing, lets find how many rows we have where all the dependent variables are null

```
x <- df_reduced[(is.na(df_reduced$property_value) &
               is.na(df_reduced$term)
               ), ]


str(x)
```

```
## 'data.frame':    12 obs. of  10 variables:
##  $ Status          : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ loan_type       : chr  "type1" "type1" "type1" "type1" ...
##  $ loan_amount     : int  106500 206500 276500 216500 146500 86500 126500 96500 196500 86500 ...
##  $ rate_of_interest: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ term            : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ property_value  : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ income          : int  3420 33540 5820 38100 4260 3780 3600 3240 5520 4620 ...
##  $ Credit_Score    : int  824 889 738 623 563 571 828 689 550 773 ...
##  $ age             : chr  "65-74" "55-64" "55-64" "65-74" ...
##  $ dtir1           : int  NA NA NA NA NA NA NA NA NA NA ...
```

Since there are 12 rows only where dependat variables are missing, lets remove those 12 variables

```
df_reduced <- df_reduced[!(is.na(df_reduced$property_value) &
               is.na(df_reduced$term)
```

```
              ), ]


str(df_reduced)
```

```
## 'data.frame':    148658 obs. of  10 variables:
##  $ Status          : int  1 1 0 0 0 0 0 0 0 0 ...
##  $ loan_type       : chr  "type1" "type2" "type1" "type1" ...
##  $ loan_amount     : int  116500 206500 406500 456500 696500 706500 346500 266500 376500 436500 ...
##  $ rate_of_interest: num  NA NA 4.56 4.25 4 ...
##  $ term            : int  360 360 360 360 360 360 360 360 360 360 ...
##  $ property_value  : int  118000 NA 508000 658000 758000 1008000 438000 308000 478000 688000 ...
##  $ income          : int  1740 4980 9480 11880 10440 10080 5040 3780 5580 6720 ...
##  $ Credit_Score    : int  758 552 834 587 602 864 860 863 580 788 ...
##  $ age             : chr  "25-34" "55-64" "35-44" "45-54" ...
##  $ dtir1           : int  45 NA 46 42 39 40 44 42 44 30 ...
```

#lets rexamine missing values for the variables of our interest

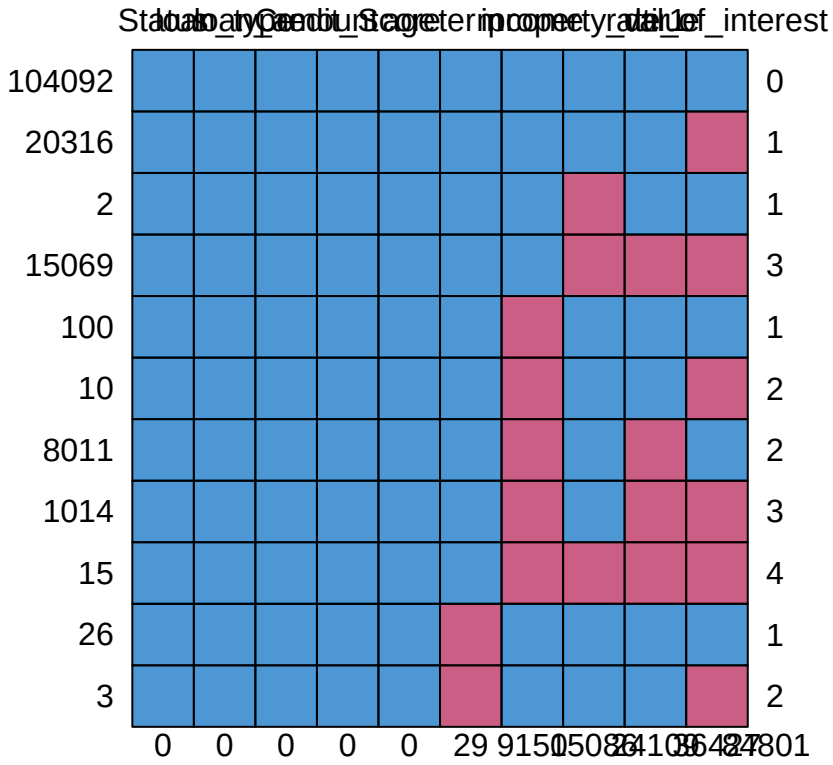
```
missing <-sapply(df_reduced, function(y) sum(length(which(is.na(y)))))
missing <- data.frame(missing)
missing
```

```
##                  missing
## Status                 0
## loan_type              0
## loan_amount            0
## rate_of_interest   36427
## term                  29
## property_value     15086
## income              9150
## Credit_Score           0
## age                    0
## dtir1              24109
```

#lets impute the values for missing variables #lets impute the misisng values based on other variables impute property_value based on loan_amount and loan_type impute term based on loan_amount and loan_type impute rate_of_interest based on loan_amount, loan_type, Term impute income based on Age impute dtir1 based property_value, loan_amount and loan_type

#lets see missing patterns

```
md.pattern(df_reduced)
```




```
##        Status loan_type loan_amount Credit_Score age term income property_value
## 104092      1         1           1            1   1    1      1              1
## 20316       1         1           1            1   1    1      1              1
## 2           1         1           1            1   1    1      1              0
## 15069       1         1           1            1   1    1      1              0
## 100         1         1           1            1   1    1      0              1
## 10          1         1           1            1   1    1      0              1
## 8011        1         1           1            1   1    1      0              1
## 1014        1         1           1            1   1    1      0              1
## 15          1         1           1            1   1    1      0              0
## 26          1         1           1            1   1    0      1              1
## 3           1         1           1            1   1    0      1              1
##             0         0           0            0   0   29   9150          15086
##        dtir1 rate_of_interest
## 104092     1                1     0
## 20316      1                0     1
## 2          1                1     1
## 15069      0                0     3
## 100        1                1     1
## 10         1                0     2
## 8011       0                1     2
## 1014       0                0     3
## 15         0                0     4
## 26         1                1     1
## 3          1                0     2
##        24109            36427 84801
```

```
#lets use mice library to do our imputation
library(mice)
init <- mice(df_reduced, meth="mean", maxit=0)
```

```
## Warning: Number of logged events: 2
```

```
#the variables we will use for imputation
init$predictorMatrix[, c("loan_amount", "loan_type")]=0
imputation <- mice(df_reduced, method=init$method,
                           predictorMatrix=init$predictorMatrix,
                           maxit=5,
                           m = 5,
                           seed=123)
```

```
##
##  iter imp variable
##   1   1  rate_of_interest  term  property_value  income  dtir1
##   1   2  rate_of_interest  term  property_value  income  dtir1
##   1   3  rate_of_interest  term  property_value  income  dtir1
##   1   4  rate_of_interest  term  property_value  income  dtir1
##   1   5  rate_of_interest  term  property_value  income  dtir1
##   2   1  rate_of_interest  term  property_value  income  dtir1
##   2   2  rate_of_interest  term  property_value  income  dtir1
##   2   3  rate_of_interest  term  property_value  income  dtir1
##   2   4  rate_of_interest  term  property_value  income  dtir1
##   2   5  rate_of_interest  term  property_value  income  dtir1
##   3   1  rate_of_interest  term  property_value  income  dtir1
##   3   2  rate_of_interest  term  property_value  income  dtir1
##   3   3  rate_of_interest  term  property_value  income  dtir1
##   3   4  rate_of_interest  term  property_value  income  dtir1
##   3   5  rate_of_interest  term  property_value  income  dtir1
##   4   1  rate_of_interest  term  property_value  income  dtir1
##   4   2  rate_of_interest  term  property_value  income  dtir1
##   4   3  rate_of_interest  term  property_value  income  dtir1
##   4   4  rate_of_interest  term  property_value  income  dtir1
##   4   5  rate_of_interest  term  property_value  income  dtir1
##   5   1  rate_of_interest  term  property_value  income  dtir1
##   5   2  rate_of_interest  term  property_value  income  dtir1
##   5   3  rate_of_interest  term  property_value  income  dtir1
##   5   4  rate_of_interest  term  property_value  income  dtir1
##   5   5  rate_of_interest  term  property_value  income  dtir1
```

```
summary(imputation)
```

```
## Class: mids
## Number of multiple imputations:  5
## Imputation methods:
##           Status        loan_type      loan_amount rate_of_interest
##               ""               ""               ""           "mean"
##             term   property_value           income     Credit_Score
##           "mean"           "mean"           "mean"               ""
##              age            dtir1
##               ""           "mean"
## PredictorMatrix:
##                   Status loan_type loan_amount rate_of_interest term
## Status                 0         0           0                1    1
## loan_type              1         0           0                1    1
## loan_amount            1         0           0                1    1
## rate_of_interest       1         0           0                0    1
## term                   1         0           0                1    0
```

```
## property_value        1         0           0               1    1
##                  property_value income Credit_Score age dtir1
## Status                        1      1            1   0     1
## loan_type                     1      1            1   0     1
## loan_amount                   1      1            1   0     1
## rate_of_interest              1      1            1   0     1
## term                          1      1            1   0     1
## property_value                0      1            1   0     1
```

#lets exmapme imputed data

```
imputation$imp$term
```

```
##               1        2        3        4        5
## 2229   335.1366 335.1366 335.1366 335.1366 335.1366
## 4374   335.1366 335.1366 335.1366 335.1366 335.1366
## 4631   335.1366 335.1366 335.1366 335.1366 335.1366
## 15167  335.1366 335.1366 335.1366 335.1366 335.1366
## 19105  335.1366 335.1366 335.1366 335.1366 335.1366
## 28251  335.1366 335.1366 335.1366 335.1366 335.1366
## 32435  335.1366 335.1366 335.1366 335.1366 335.1366
## 41304  335.1366 335.1366 335.1366 335.1366 335.1366
## 48176  335.1366 335.1366 335.1366 335.1366 335.1366
## 49998  335.1366 335.1366 335.1366 335.1366 335.1366
## 50584  335.1366 335.1366 335.1366 335.1366 335.1366
## 53197  335.1366 335.1366 335.1366 335.1366 335.1366
## 62812  335.1366 335.1366 335.1366 335.1366 335.1366
## 65761  335.1366 335.1366 335.1366 335.1366 335.1366
## 66821  335.1366 335.1366 335.1366 335.1366 335.1366
## 68466  335.1366 335.1366 335.1366 335.1366 335.1366
## 73070  335.1366 335.1366 335.1366 335.1366 335.1366
## 81824  335.1366 335.1366 335.1366 335.1366 335.1366
## 81899  335.1366 335.1366 335.1366 335.1366 335.1366
## 84653  335.1366 335.1366 335.1366 335.1366 335.1366
## 90204  335.1366 335.1366 335.1366 335.1366 335.1366
## 99269  335.1366 335.1366 335.1366 335.1366 335.1366
## 108639 335.1366 335.1366 335.1366 335.1366 335.1366
## 113909 335.1366 335.1366 335.1366 335.1366 335.1366
## 117526 335.1366 335.1366 335.1366 335.1366 335.1366
## 131008 335.1366 335.1366 335.1366 335.1366 335.1366
## 131165 335.1366 335.1366 335.1366 335.1366 335.1366
## 143317 335.1366 335.1366 335.1366 335.1366 335.1366
## 146410 335.1366 335.1366 335.1366 335.1366 335.1366
```

```
head(imputation$imp$rate_of_interest)
```

```
##           1        2        3        4        5
## 1  4.045476 4.045476 4.045476 4.045476 4.045476
## 2  4.045476 4.045476 4.045476 4.045476 4.045476
## 11 4.045476 4.045476 4.045476 4.045476 4.045476
## 13 4.045476 4.045476 4.045476 4.045476 4.045476
## 16 4.045476 4.045476 4.045476 4.045476 4.045476
## 17 4.045476 4.045476 4.045476 4.045476 4.045476
```

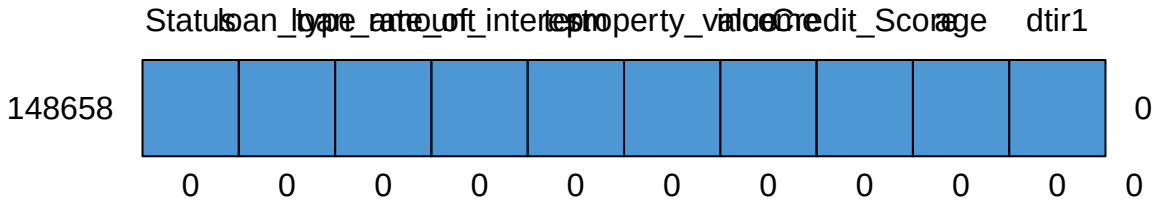
#based on the imputed data it loosk like all of the imputaion methods prodcued same data #so lets use the data from 1st imputation method

```
df_imputed <- complete(imputation,1)
```

#lets examine the imputed data

```
md.pattern(df_imputed)
```

```
##  /\     /\
## {  `---'  }
## {  O   O  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
```




```
##        Status loan_type loan_amount rate_of_interest term property_value income
## 148658      1         1           1                1    1              1      1
##             0         0           0                0    0              0      0
##        Credit_Score age dtir1
## 148658            1   1     1 0
##                   0   0     0 0
```

#lets remove the outliers

```
summary(df_imputed)
```

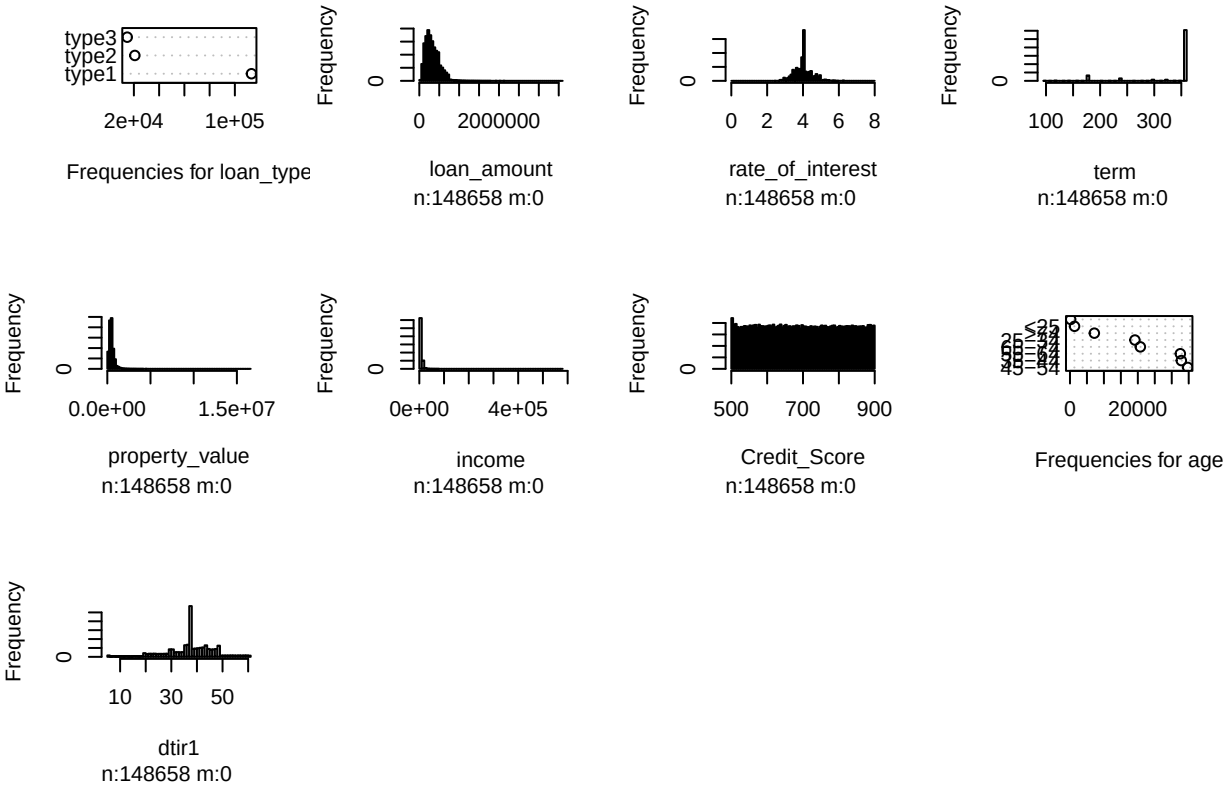
```
##      Status        loan_type          loan_amount      rate_of_interest
##  Min.   :0.0000   Length:148658      Min.   :  16500   Min.   :0.000
##  1st Qu.:0.0000   Class :character   1st Qu.: 196500   1st Qu.:3.750
##  Median :0.0000   Mode  :character   Median : 296500   Median :4.045
##  Mean   :0.2464                      Mean   : 331132   Mean   :4.045
##  3rd Qu.:0.0000                      3rd Qu.: 436500   3rd Qu.:4.250
##  Max.   :1.0000                      Max.   :3576500   Max.   :8.000
##       term       property_value         income        Credit_Score
##  Min.   : 96.0   Min.   :    8000   Min.   :     0   Min.   :500.0
##  1st Qu.:360.0   1st Qu.:  288000   1st Qu.:  3840   1st Qu.:599.0
##  Median :360.0   Median :  458000   Median :  6000   Median :699.0
##  Mean   :335.1   Mean   :  497893   Mean   :  6957   Mean   :699.8
##  3rd Qu.:360.0   3rd Qu.:  598000   3rd Qu.:  8280   3rd Qu.:800.0
##  Max.   :360.0   Max.   :16508000   Max.   :578580   Max.   :900.0
##      age                dtir1
##  Length:148658      Min.   : 5.00
##  Class :character   1st Qu.:33.00
##  Mode  :character   Median :37.73
##                     Mean   :37.73
##                     3rd Qu.:44.00
##                     Max.   :61.00
```

```
#install.packages("Hmisc")
library(Hmisc)
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'

## The following object is masked from 'package:caret':
##
##     cluster

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following object is masked from 'package:sjmisc':
##
##     %nin%

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

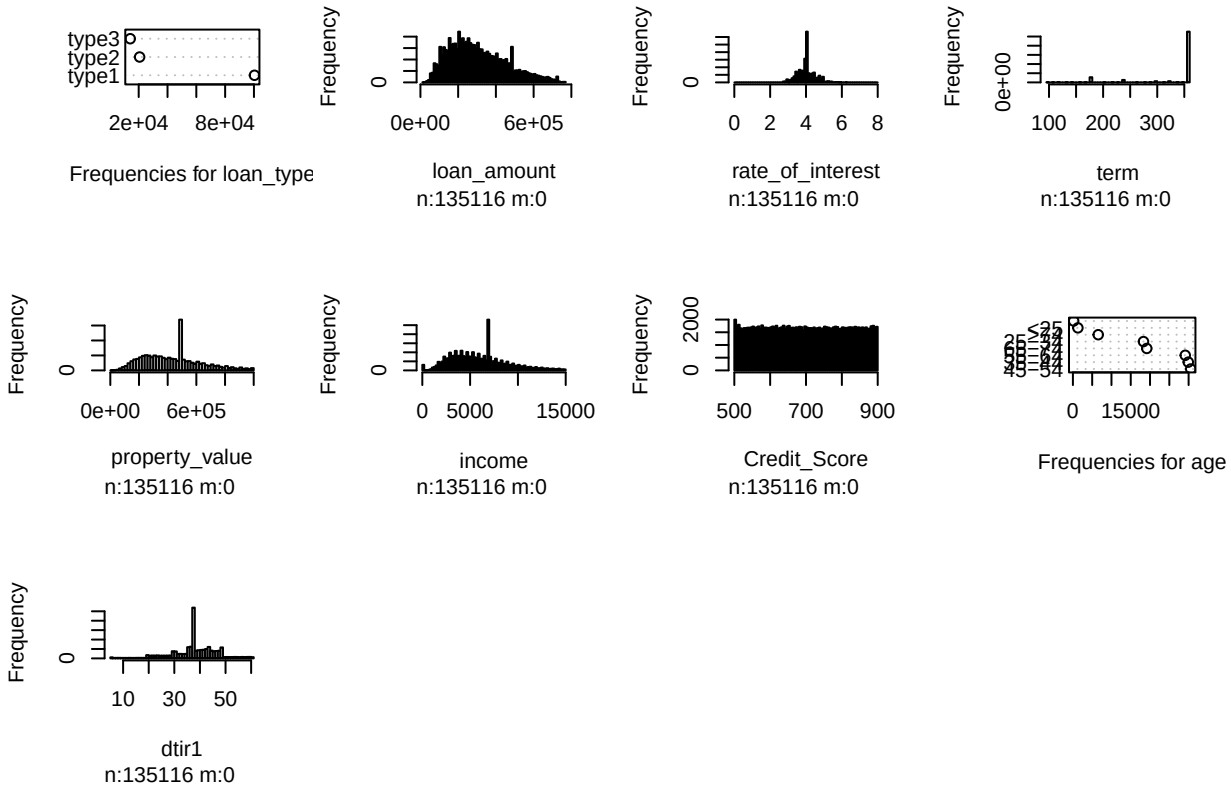
```
hist.data.frame(df_imputed)
```



#we can see that the follwing variables have outliers #loan_amount, property_value, income # we will use the formula Q3 + (1.5*IQR) to remove outlier observations

```
#income
outliers <- boxplot(df_imputed$income, plot=FALSE)$out
df_imputed<- df_imputed[-which(df_imputed$income %in% outliers),]

#loan_amount
```

```
outliers <- boxplot(df_imputed$loan_amount, plot=FALSE)$out
df_imputed<- df_imputed[-which(df_imputed$loan_amount %in% outliers),]

#property_value
outliers <- boxplot(df_imputed$property_value, plot=FALSE)$out
df_imputed<- df_imputed[-which(df_imputed$property_value %in% outliers),]

hist.data.frame(df_imputed)
```

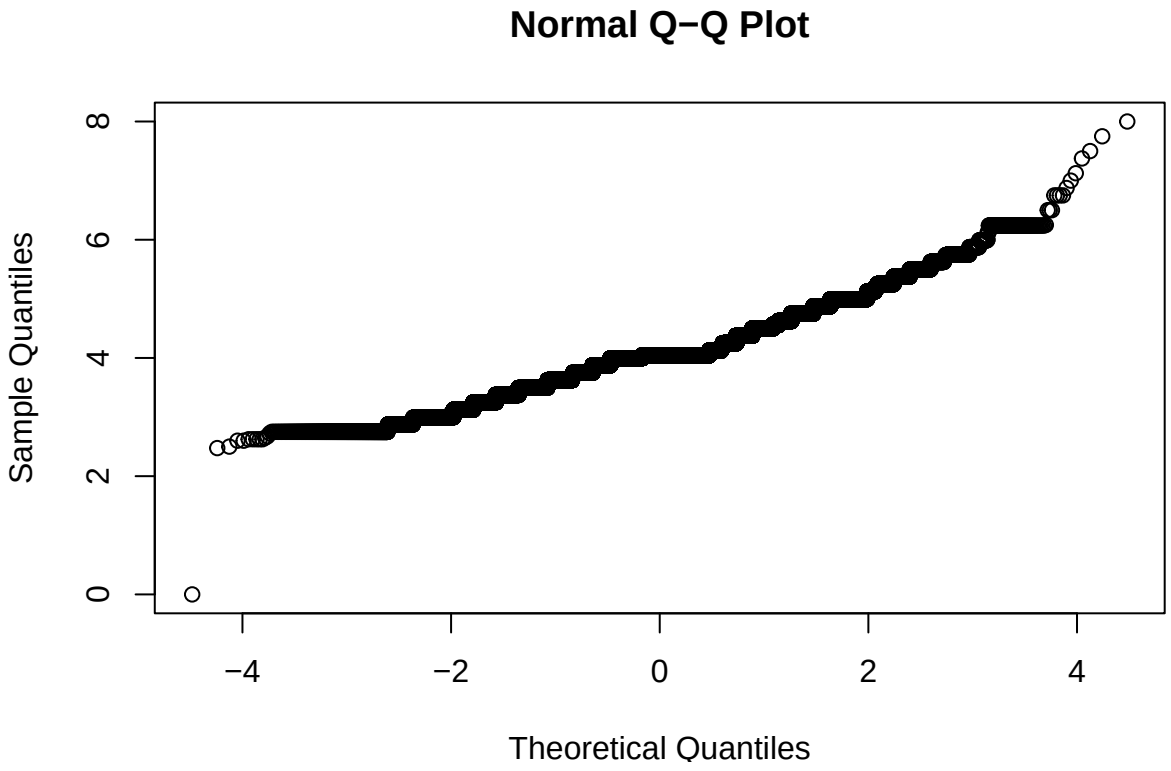



```
summary(df_imputed)
```

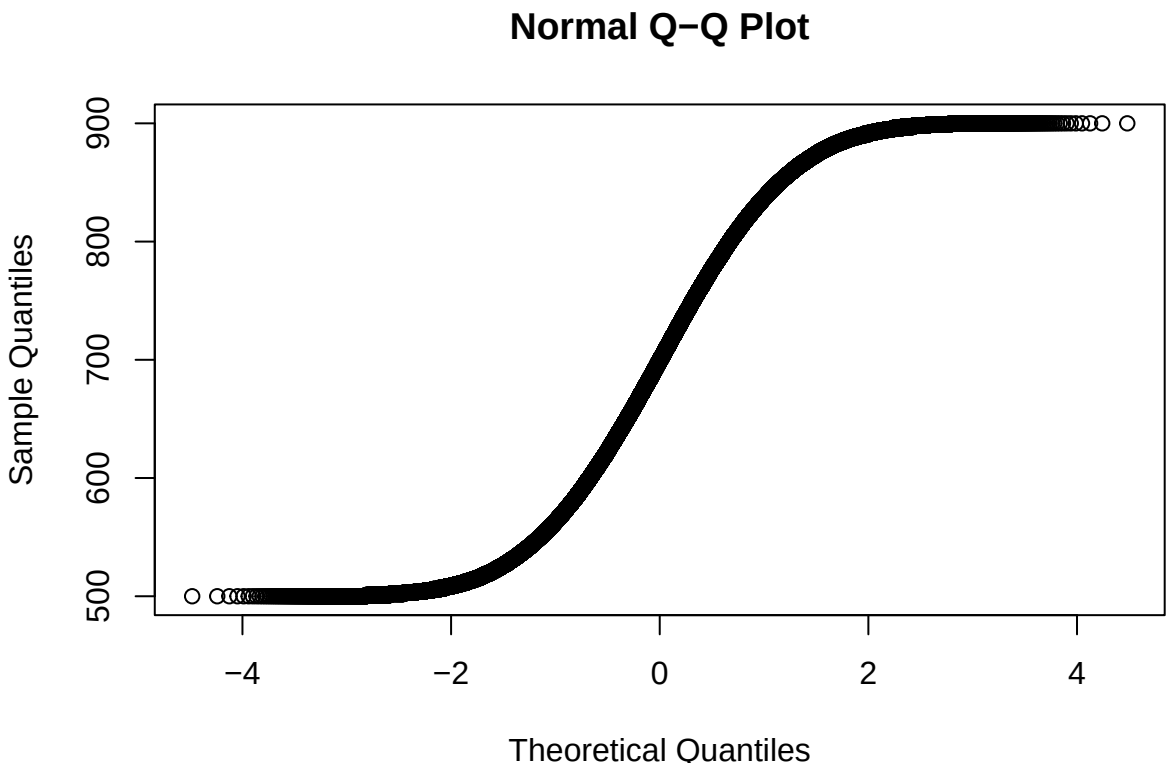
```
##      Status         loan_type          loan_amount     rate_of_interest
##  Min.   :0.0000   Length:135116      Min.   : 16500   Min.   :0.000
##  1st Qu.:0.0000   Class :character   1st Qu.:186500   1st Qu.:3.750
##  Median :0.0000   Mode  :character   Median :286500   Median :4.045
##  Mean   :0.2499                      Mean   :306515   Mean   :4.048
##  3rd Qu.:0.0000                      3rd Qu.:406500   3rd Qu.:4.250
##  Max.   :1.0000                      Max.   :766500   Max.   :8.000
##       term       property_value       income       Credit_Score
##  Min.   : 96.0   Min.   :  8000   Min.   :    0   Min.   :500.0
##  1st Qu.:360.0   1st Qu.:278000   1st Qu.: 3660   1st Qu.:599.0
##  Median :360.0   Median :428000   Median : 5640   Median :699.0
##  Mean   :335.6   Mean   :435352   Mean   : 5944   Mean   :699.8
##  3rd Qu.:360.0   3rd Qu.:548000   3rd Qu.: 7500   3rd Qu.:800.0
##  Max.   :360.0   Max.   :998000   Max.   :14940   Max.   :900.0
##      age                dtir1
##  Length:135116      Min.   : 5.00
##  Class :character   1st Qu.:34.00
##  Mode  :character   Median :37.73
```

```
##                     Mean   :38.16
##                     3rd Qu.:44.00
##                     Max.   :61.00
```

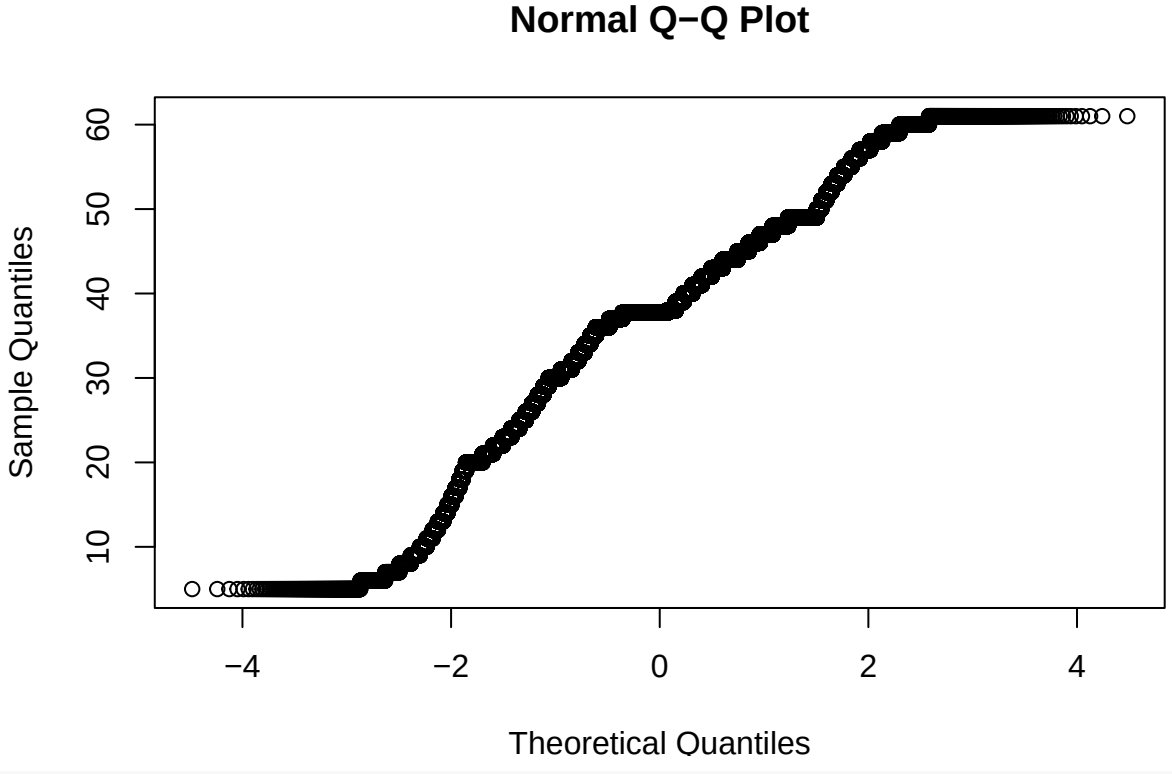
```
qqnorm(df_imputed$rate_of_interest)
```



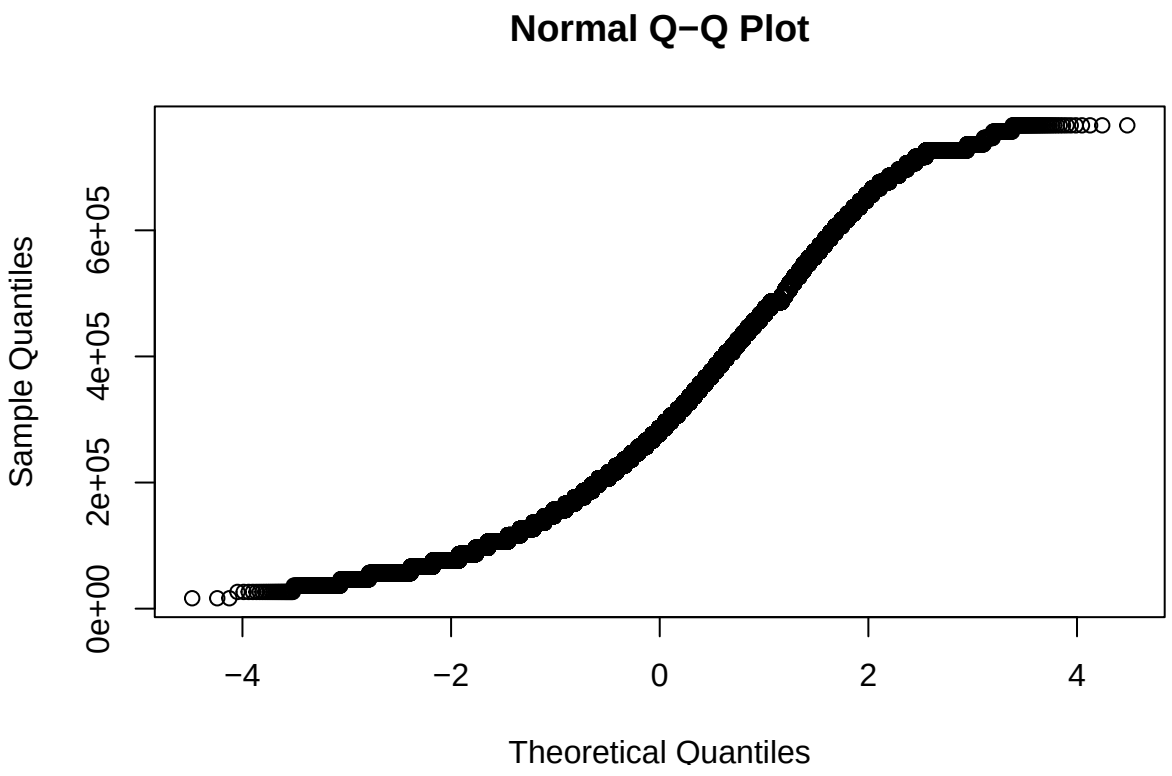

```
qqnorm(df_imputed$Credit_Score)
```



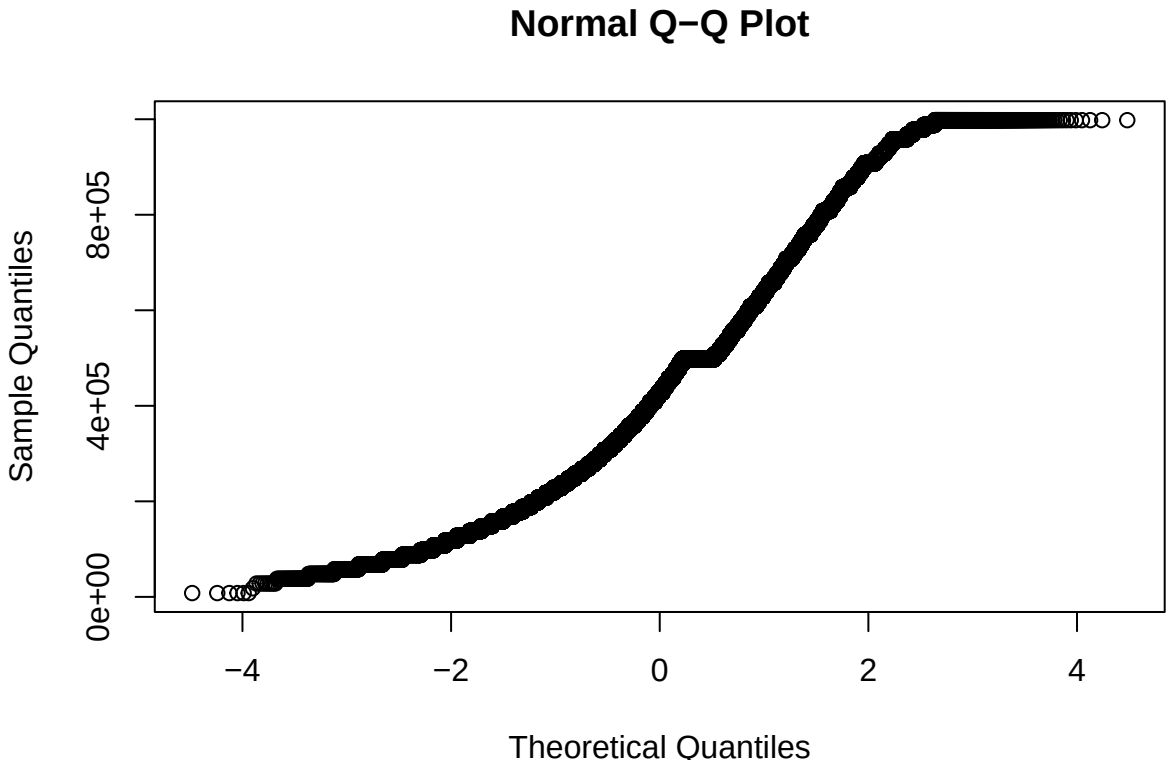

```
qqnorm(df_imputed$dtir1)
```




```
qqnorm(df_imputed$loan_amount )
```




```
qqnorm(df_imputed$ property_value)
```



##Create co-relation plots of our data #lets factorize the categorical data

```
df_imputed$loan_type <- factor(df_imputed$loan_type)
df_imputed$age <- factor(df_imputed$age)
```

```
summary(df_imputed)
```

```
##      Status       loan_type       loan_amount     rate_of_interest
##  Min.   :0.0000   type1:100363   Min.   : 16500   Min.   :0.000
##  1st Qu.:0.0000   type2: 20629   1st Qu.:186500   1st Qu.:3.750
##  Median :0.0000   type3: 14124   Median :286500   Median :4.045
##  Mean   :0.2499                  Mean   :306515   Mean   :4.048
##  3rd Qu.:0.0000                  3rd Qu.:406500   3rd Qu.:4.250
##  Max.   :1.0000                  Max.   :766500   Max.   :8.000
##
##       term       property_value       income       Credit_Score
##  Min.   : 96.0   Min.   :  8000   Min.   :    0   Min.   :500.0
##  1st Qu.:360.0   1st Qu.:278000   1st Qu.: 3660   1st Qu.:599.0
##  Median :360.0   Median :428000   Median : 5640   Median :699.0
##  Mean   :335.6   Mean   :435352   Mean   : 5944   Mean   :699.8
##  3rd Qu.:360.0   3rd Qu.:548000   3rd Qu.: 7500   3rd Qu.:800.0
##  Max.   :360.0   Max.   :998000   Max.   :14940   Max.   :900.0
##
##       age            dtir1
##  45-54  :30559   Min.   : 5.00
##  35-44  :29916   1st Qu.:34.00
##  55-64  :29131   Median :37.73
##  65-74  :19155   Mean   :38.16
##  25-34  :18298   3rd Qu.:44.00
##  >74    : 6565   Max.   :61.00
##  (Other): 1492
```

#lets do a glm on the data before cleanup to identify if it works

```
model <- glm(Status ~ loan_type+loan_amount+rate_of_interest+term+property_value+income+Credit_Score+ag
summary(model)
```

```
##
## Call:
## glm(formula = Status ~ loan_type + loan_amount + rate_of_interest +
##     term + property_value + income + Credit_Score + age + dtir1,
##     family = binomial(link = "logit"), data = df_imputed)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3956  -0.7875  -0.6853  -0.3822   2.3770
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       1.393e+01  3.776e+01   0.369   0.7121
## loan_typetype2    5.353e-01  1.817e-02  29.469  < 2e-16 ***
## loan_typetype3    1.445e-01  2.273e-02   6.359 2.03e-10 ***
## loan_amount      -4.296e-07  8.138e-08  -5.279 1.30e-07 ***
## rate_of_interest  2.441e-02  1.469e-02   1.661   0.0967 .
## term             -7.970e-04  1.214e-04  -6.567 5.14e-11 ***
## property_value    1.067e-06  5.300e-08  20.139  < 2e-16 ***
## income           -1.304e-04  3.120e-06 -41.811  < 2e-16 ***
## Credit_Score      7.222e-05  5.520e-05   1.308   0.1908
## age<25           -1.461e+01  3.776e+01  -0.387   0.6989
## age>74           -1.456e+01  3.776e+01  -0.386   0.6998
```

```
## age25-34         -1.477e+01  3.776e+01  -0.391   0.6956
## age35-44         -1.472e+01  3.776e+01  -0.390   0.6967
## age45-54         -1.464e+01  3.776e+01  -0.388   0.6983
## age55-64         -1.461e+01  3.776e+01  -0.387   0.6988
## age65-74         -1.466e+01  3.776e+01  -0.388   0.6978
## dtir1             1.249e-03  7.397e-04   1.689   0.0912 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 151921  on 135115  degrees of freedom
## Residual deviance: 147146  on 135099  degrees of freedom
## AIC: 147180
##
## Number of Fisher Scoring iterations: 12
```

With the given variables above, the data are classified as follows:

VARIABLE CATEGORY SCALE OF MEASUREMENT Status Categorical Nominal Loan_type Categorical Nominal Loan_Amount Numerical Continuous Rate_of_interest Numerical Continuous Term Numerical Discrete Property_value Numerical Continuous Income Numerical Continuous Credit_Score Numerical Continuous Age Categorical Ordinal Dtir1 (Debt to Income Ratio) Numerical Continuous

#These variables are then checked for association. Categorical variables underwent a chi-square test of independence to #check if these are significantly associated or not. The following hypotheses were developed:

Null Hypothesis (Ho) - The variables are not related to the population. Alternative Hypothesis (Ha) - The variables are related to the population. # Here are the results of the chi-square test:

```
# For Status and Loan Type
Cat1 = table(df_imputed$Status, df_imputed$loan_type)
Cat1
```

```
##
##     type1 type2 type3
##   0 77285 13500 10570
##   1 23078  7129  3554
```

```
chisq.test(df_imputed$Status, df_imputed$loan_type,correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  df_imputed$Status and df_imputed$loan_type
## X-squared = 1221, df = 2, p-value < 2.2e-16
```

```
# For Status and Age
Cat2 = table(df_imputed$Status, df_imputed$age)
Cat2
```

```
##
##             <25   >74 25-34 35-44 45-54 55-64 65-74
##   0     0   918  4566 14233 23207 23050 21439 13942
##   1   199   375  1999  4065  6709  7509  7692  5213
```

```
chisq.test(df_imputed$Status, df_imputed$age,correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  df_imputed$Status and df_imputed$age
## X-squared = 977.48, df = 7, p-value < 2.2e-16
```

Based on the p-values, we can conclude that status and loan type, as well as age are related

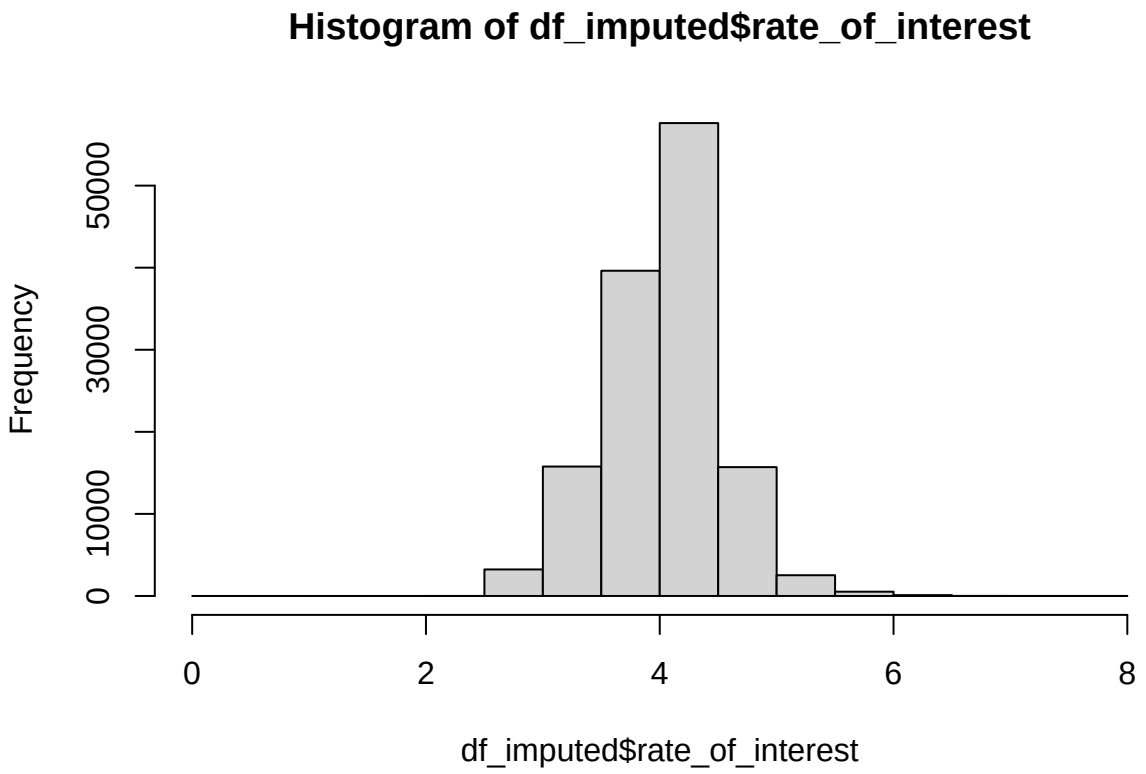
and there is a proven association.

For numerical variables, we are doing a two sample t-test, as status only has 2 categories (Approved or Declined). However, there are certain assumptions to be met: 1. The data should be continuous 2. The data should follow a normal bell-shaped curve visually (NOTE: I am not sure if this is relevant since there is also an automatic assumption of normality from Measures of Central Tendency should the data be large) 3. The data must come from a random sample 4. The variances for the 2 groups are equal 5. There should be a sizable amount of data

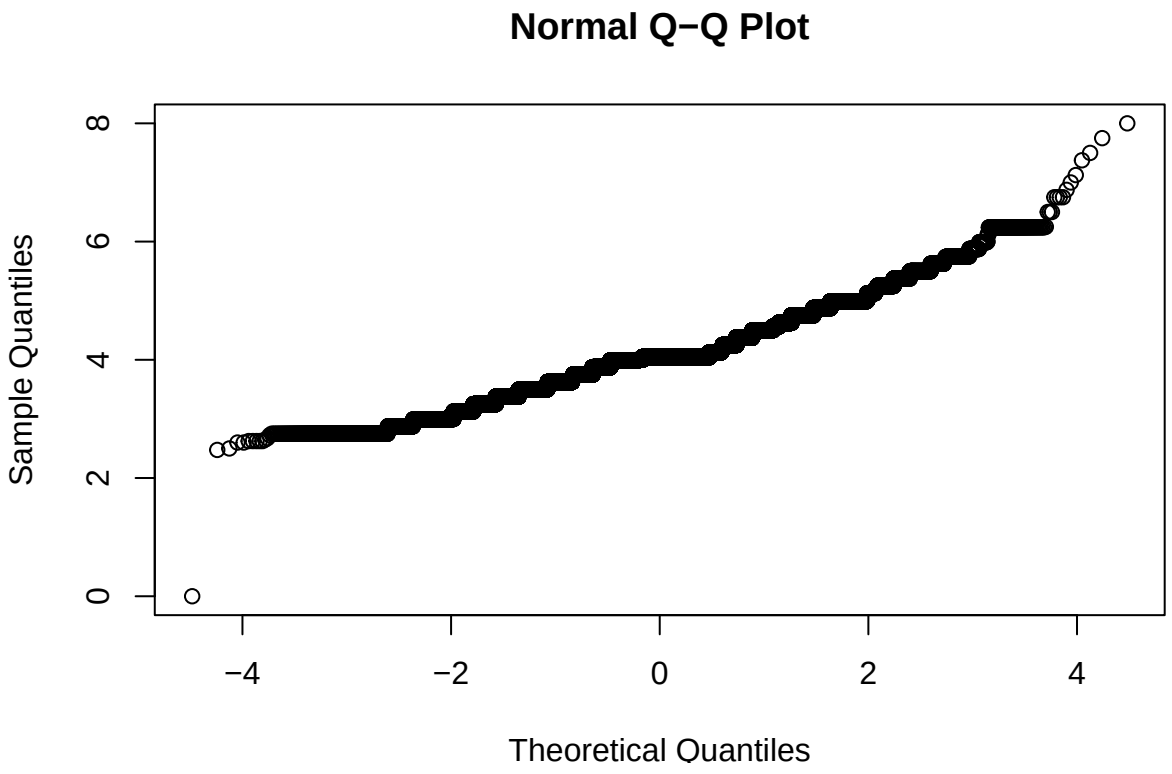
These are the variables with Normal Distribution

(this is not important)

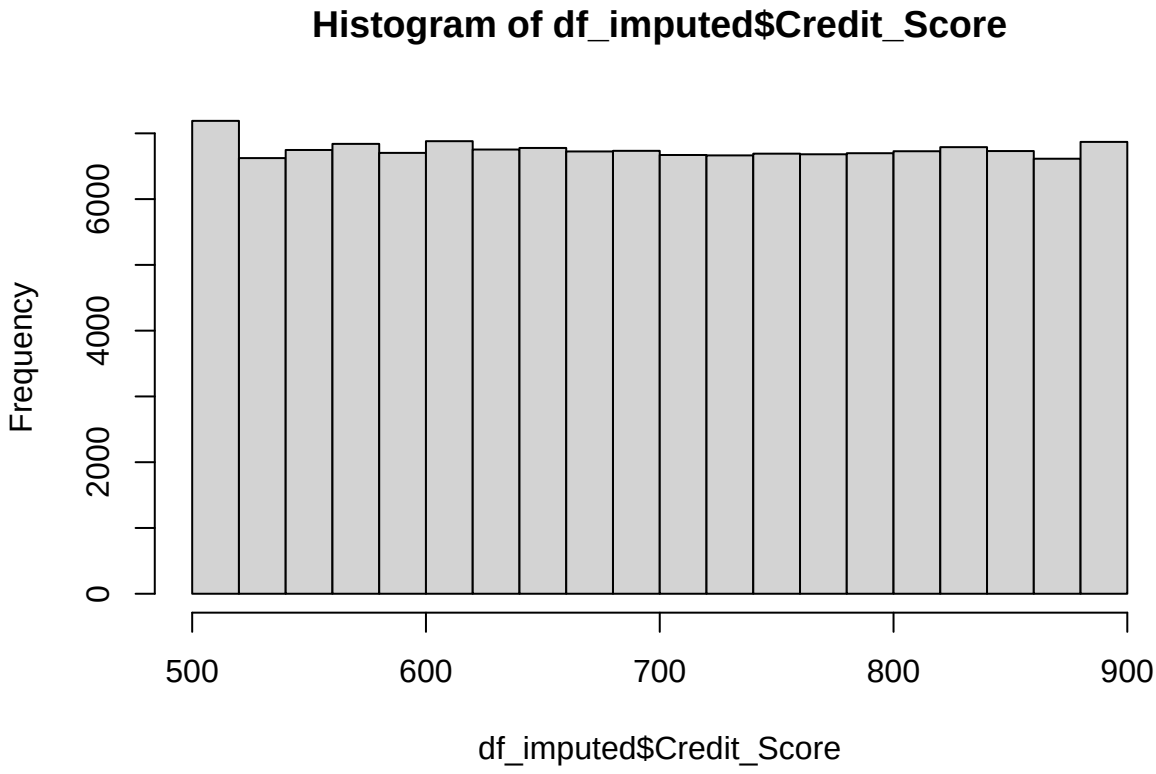
```
hist(df_imputed$rate_of_interest)
```



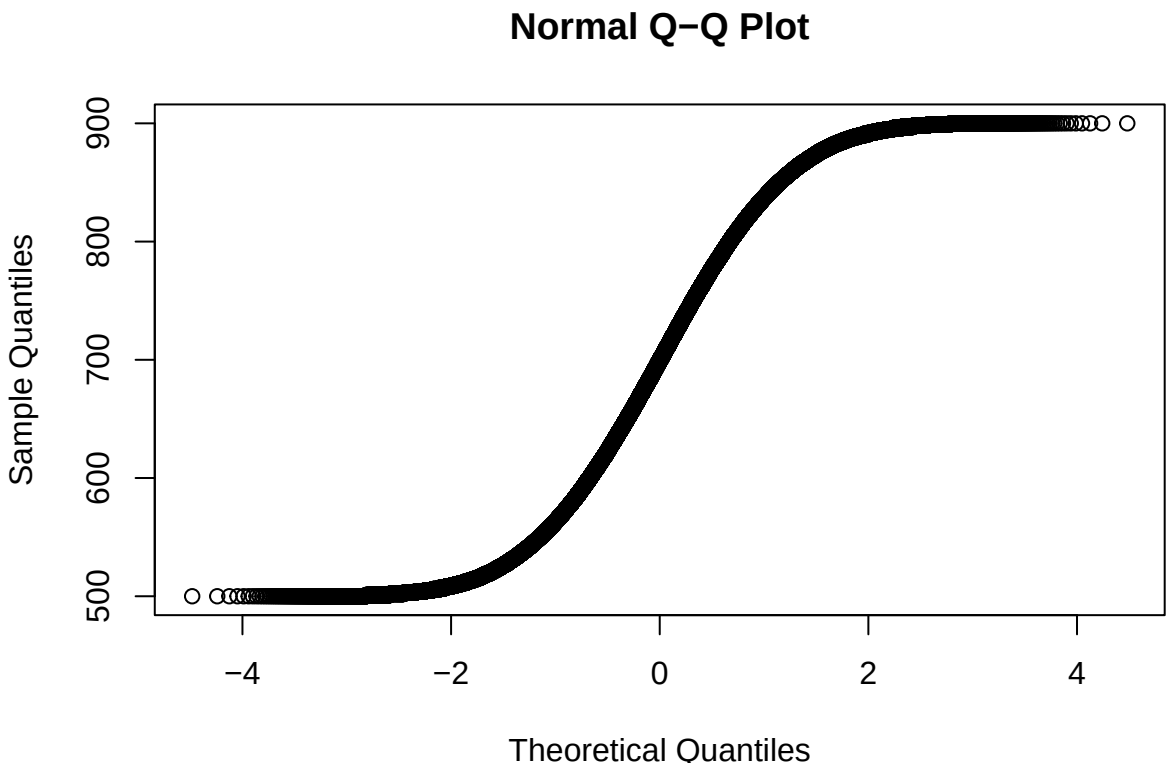

```
qqnorm(df_imputed$rate_of_interest)
```



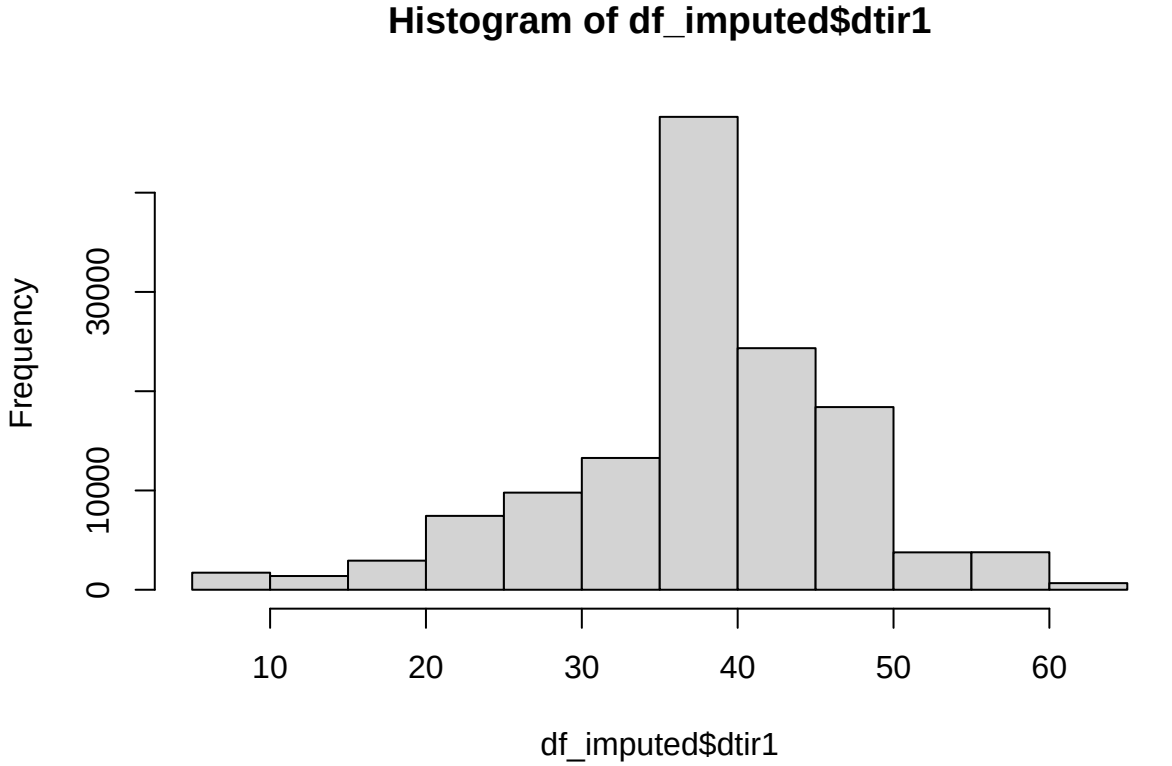

```
hist(df_imputed$Credit_Score)
```



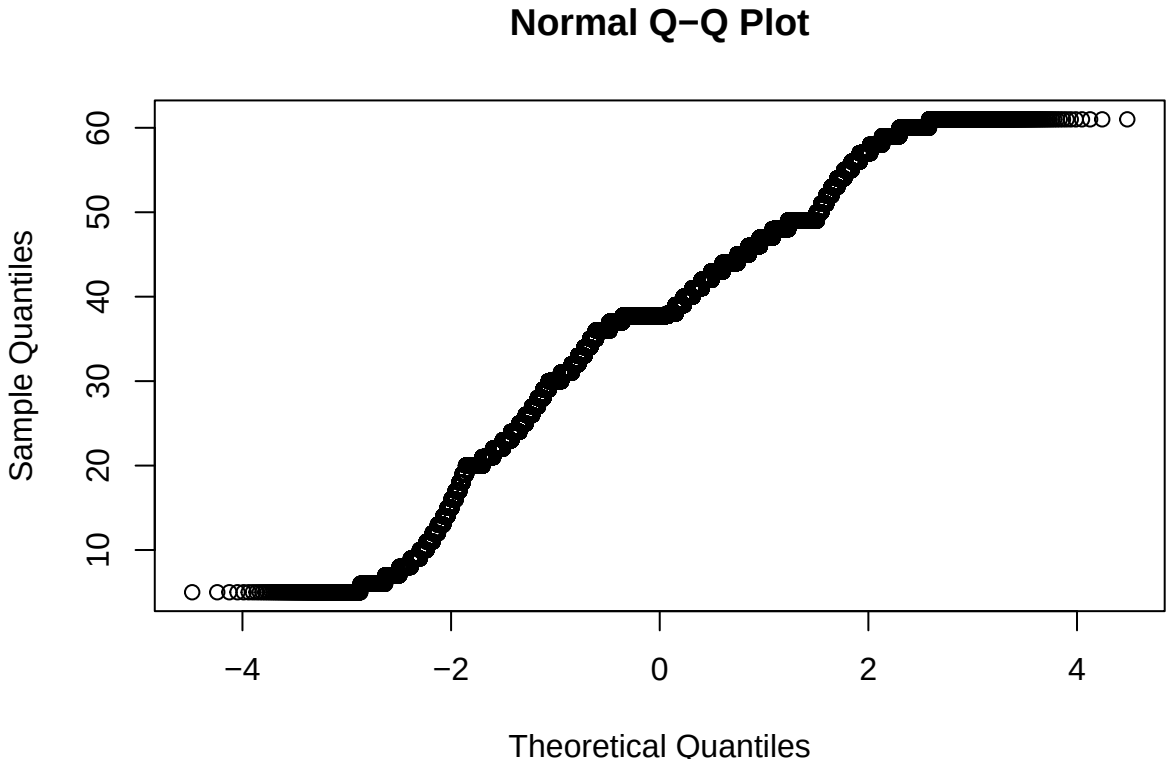

```
qqnorm(df_imputed$Credit_Score)
```




```
hist(df_imputed$dtir1)
```




```
qqnorm(df_imputed$dtir1)
```



For those variables that followed a visually normal distribution, a test was done to check whether the # variances are equal or not. Here are the results assuming a level of significance = 0.05:

```
imputed_data <- df_imputed
# For status and rate of interest
var.test(imputed_data$Status,imputed_data$rate_of_interest, alternative="two.sided",conf.level=0.95)
```

```
##
##  F test to compare two variances
##
## data:  imputed_data$Status and imputed_data$rate_of_interest
## F = 0.81697, num df = 135115, denom df = 135115, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.8083081 0.8257331
## sample estimates:
## ratio of variances
##          0.8169741
```

```
# The test above states that variances are not equal - so we conduct a t-test assuming unequal variance
t.test(imputed_data$Status,imputed_data$rate_of_interest, alternative="two.sided", var.equal = FALSE,co
```

```
##
##  Welch Two Sample t-test
##
## data:  imputed_data$Status and imputed_data$rate_of_interest
## t = -2162.3, df = 267516, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.801446 -3.794561
## sample estimates:
## mean of x mean of y
```

```
## 0.2498668 4.0478699
```

```
# For status and debt to income ratio
var.test(imputed_data$Status,imputed_data$dtir1, alternative="two.sided",conf.level=0.95)
```

```
##
##  F test to compare two variances
##
## data:  imputed_data$Status and imputed_data$dtir1
## F = 0.0021566, num df = 135115, denom df = 135115, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.002133736 0.002179734
## sample estimates:
## ratio of variances
##        0.002156612
```

```
# The test above states that variances are not equal - so we conduct a t-test assuming unequal variance
t.test(imputed_data$Status,imputed_data$dtir1, alternative="two.sided", var.equal = FALSE,conf.level = 
```

```
##
##  Welch Two Sample t-test
##
## data:  imputed_data$Status and imputed_data$dtir1
## t = -1493.2, df = 135698, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -37.96178 -37.86225
## sample estimates:
##  mean of x  mean of y
##  0.2498668 38.1618827
```

```
# For status and Credit
var.test(imputed_data$Status,imputed_data$Credit, alternative="two.sided",conf.level=0.95)
```

```
##
##  F test to compare two variances
##
## data:  imputed_data$Status and imputed_data$Credit
## F = 1.396e-05, num df = 135115, denom df = 135115, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.381155e-05 1.410929e-05
## sample estimates:
## ratio of variances
##       1.395962e-05
```

```
# The test above states that variances are not equal - so we conduct a t-test assuming unequal variance
t.test(imputed_data$Status,imputed_data$Credit, alternative="two.sided", var.equal = FALSE,conf.level =
```

```
##
##  Welch Two Sample t-test
##
## data:  imputed_data$Status and imputed_data$Credit
## t = -2219, df = 135119, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  -700.1333 -698.8976
## sample estimates:
##   mean of x   mean of y
##   0.2498668 699.7653276
```

Basing on the above t-tests, we can reject Ho and prove that there is association between the numerical variables above and loan status.

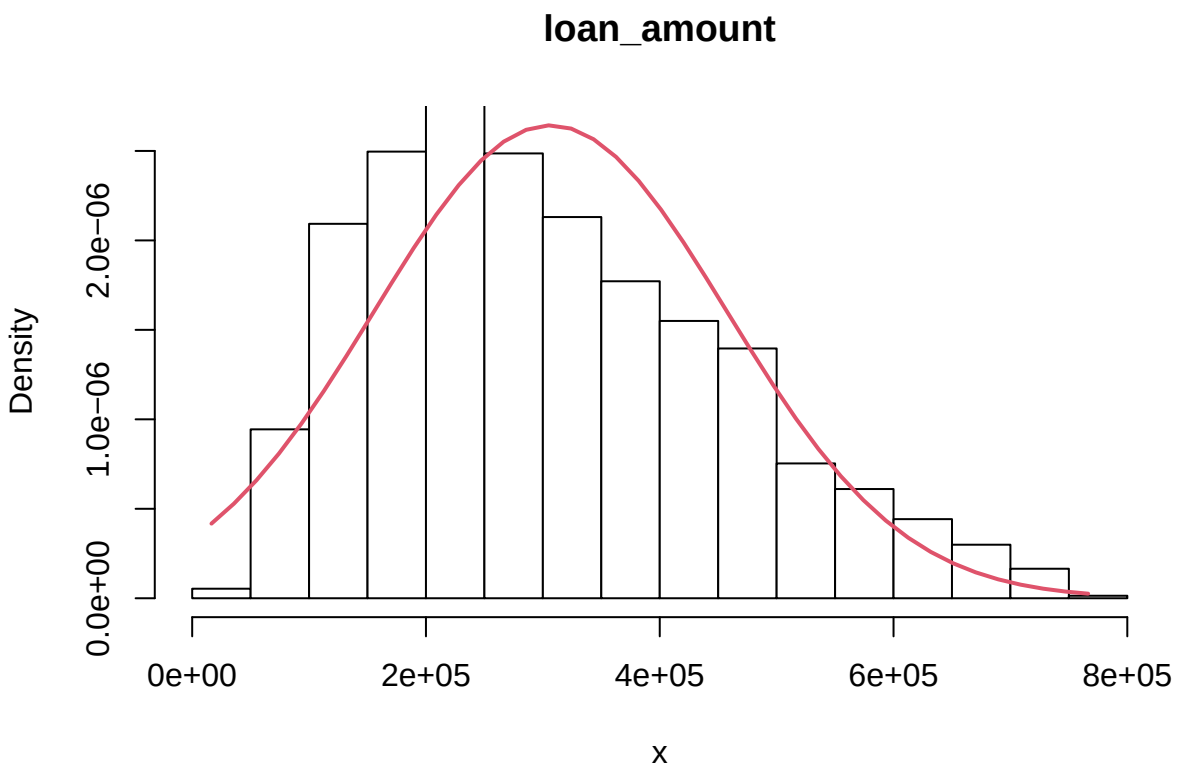
These are the numerical variables which do not follow a normal distribution. Skewness is also added for transformation purposes, as necessary.

```
library(ggpubr)
library(moments)
```

```
##
## Attaching package: 'moments'
```

```
## The following objects are masked from 'package:PerformanceAnalytics':
##
##     kurtosis, skewness
```

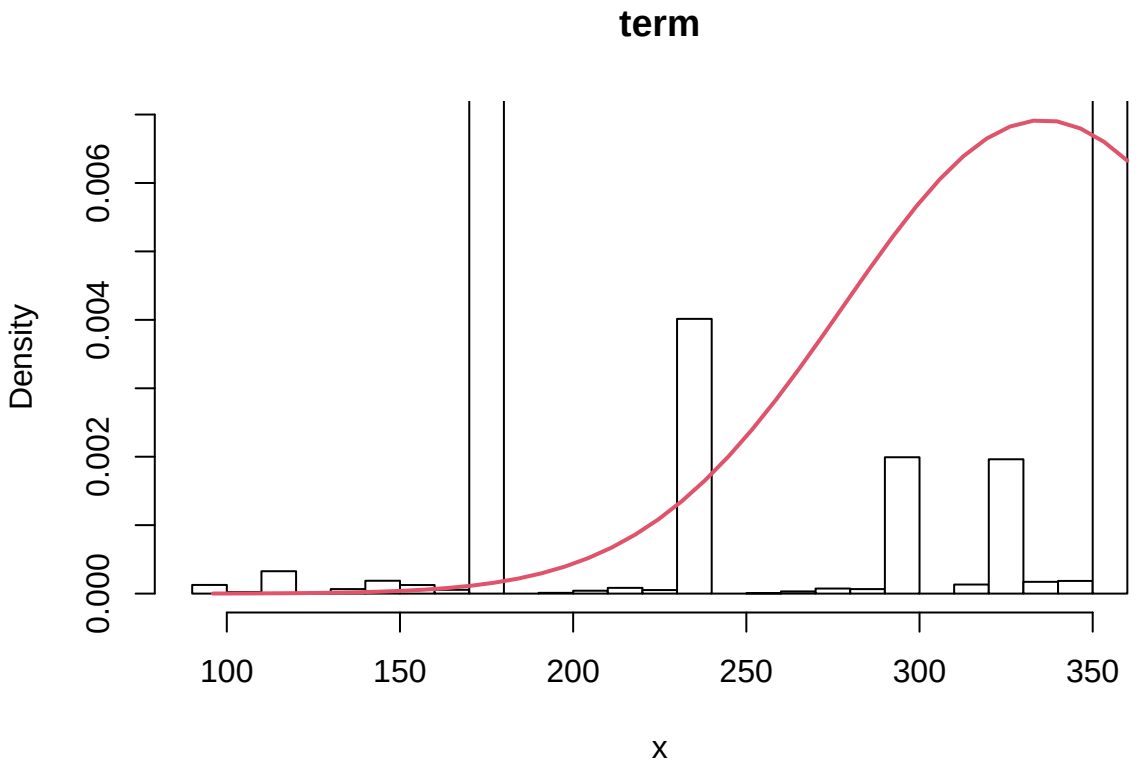
```
# imputed_data$loan_amount
x <- imputed_data$loan_amount
x2 <- seq(min(x), max(x), length = 40)
# Normal curve
fun <- dnorm(x2, mean = mean(x), sd = sd(x))
# Histogram
hist(x, prob = TRUE, col = "white",
     ylim = c(0, max(fun)),
     main = "loan_amount")
lines(x2, fun, col = 2, lwd = 2)
```




```
skewness(imputed_data$loan_amount, na.rm = TRUE)
```

```
## [1] 0.5892285
```

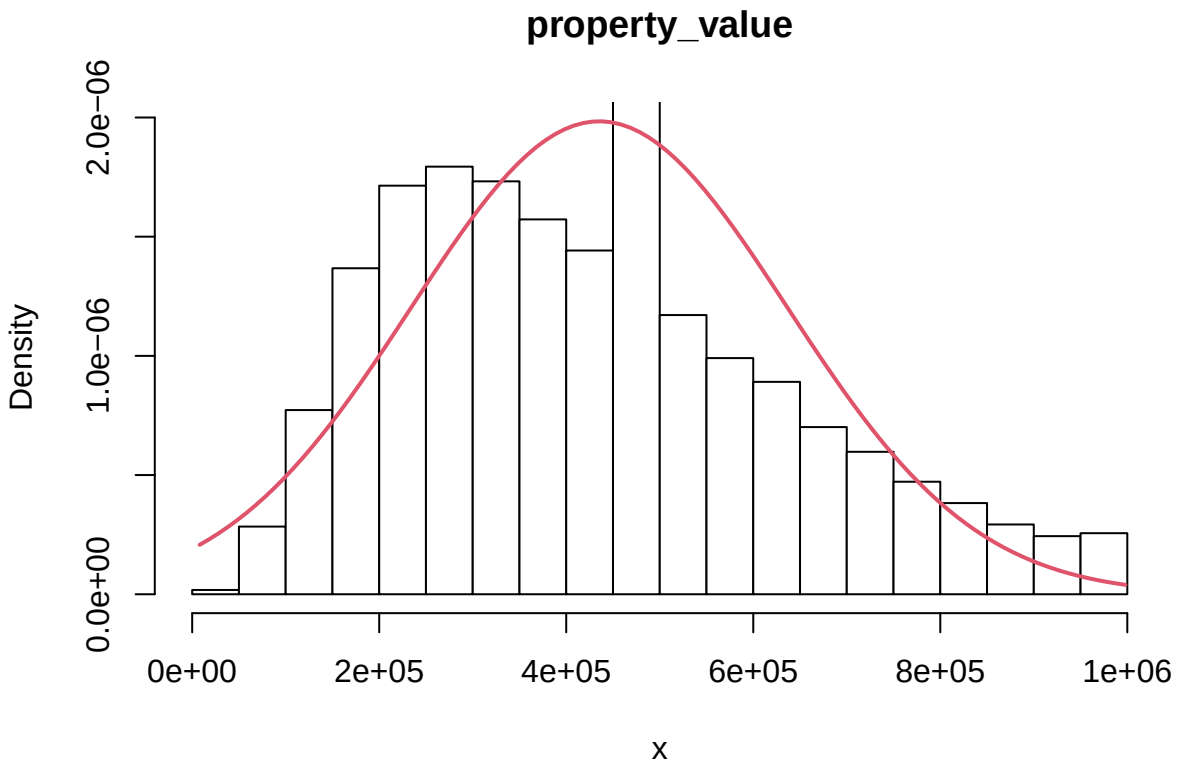
```
# imputed_data$term
x <- imputed_data$term
x2 <- seq(min(x), max(x), length = 40)
# Normal curve
fun <- dnorm(x2, mean = mean(x), sd = sd(x))
# Histogram
hist(x, prob = TRUE, col = "white",
     ylim = c(0, max(fun)),
     main = "term")
lines(x2, fun, col = 2, lwd = 2)
```




```
skewness(imputed_data$term, na.rm = TRUE)
```

```
## [1] -2.210159
```

```
# imputed_data$property_value
x <- imputed_data$property_value
x2 <- seq(min(x), max(x), length = 100)
# Normal curve
fun <- dnorm(x2, mean = mean(x), sd = sd(x))
# Histogram
hist(x, prob = TRUE, col = "white",
     ylim = c(0, max(fun)),
     main = "property_value")
lines(x2, fun, col = 2, lwd = 2)
```




```
skewness(imputed_data$property_value, na.rm = TRUE)
```

```
## [1] 0.5217276
```

```
# imputed_data$property_value
x <- imputed_data$income
x2 <- seq(min(x), max(x), length = 40)
# Normal curve
fun <- dnorm(x2, mean = mean(x), sd = sd(x))
# Histogram
hist(x, prob = TRUE, col = "white",
     ylim = c(0, max(fun)),
     main = "income")
lines(x2, fun, col = 2, lwd = 2)
```


```
skewness(imputed_data$income, na.rm = TRUE)
```

```
## [1] 0.6172749
```

Furthermore, a Kolmogorov-Smirnov Test was done with the following hypothesis:

Null Hypothesis (Ho) - Data is normally distributed Alternative Hypothesis (Ha) - Data is not normally distributed

Here are the results:

```
# Status and Loan Amount
ks.test(imputed_data$loan_amount, 'pnorm') # Not normal distribution (make this log)
```

```
## Warning in ks.test.default(imputed_data$loan_amount, "pnorm"): ties should not
## be present for the Kolmogorov-Smirnov test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  imputed_data$loan_amount
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
ks.test(imputed_data$term, 'pnorm') # Not normal distribution (make this log)
```

```
## Warning in ks.test.default(imputed_data$term, "pnorm"): ties should not be
## present for the Kolmogorov-Smirnov test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
```

```
##
## data:  imputed_data$term
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
ks.test(imputed_data$property_value,  'pnorm') # Not normal distribution (make this log)
```

```
## Warning in ks.test.default(imputed_data$property_value, "pnorm"): ties should
## not be present for the Kolmogorov-Smirnov test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  imputed_data$property_value
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
ks.test(imputed_data$income,  'pnorm') # Not normal distribution (make this log)
```

```
## Warning in ks.test.default(imputed_data$income, "pnorm"): ties should not be
## present for the Kolmogorov-Smirnov test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  imputed_data$income
## D = 0.99106, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Running the GLM Model without interactions

Checking for Multicollinearity

```
library(car)
library(olsrr)

model <- glm(Status ~ loan_type+rate_of_interest+term+income+Credit_Score+age+dtir1+loan_amount+property
summary(model)
```

```
##
## Call:
## glm(formula = Status ~ loan_type + rate_of_interest + term +
##     income + Credit_Score + age + dtir1 + loan_amount + property_value,
##     family = binomial(link = "logit"), data = imputed_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3956  -0.7875  -0.6853  -0.3822   2.3770
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       1.393e+01  3.776e+01   0.369   0.7121
## loan_typetype2    5.353e-01  1.817e-02  29.469  < 2e-16 ***
## loan_typetype3    1.445e-01  2.273e-02   6.359 2.03e-10 ***
## rate_of_interest  2.441e-02  1.469e-02   1.661   0.0967 .
```

```
## term             -7.970e-04  1.214e-04  -6.567 5.14e-11 ***
## income           -1.304e-04  3.120e-06 -41.811  < 2e-16 ***
## Credit_Score      7.222e-05  5.520e-05   1.308   0.1908
## age<25           -1.461e+01  3.776e+01  -0.387   0.6989
## age>74           -1.456e+01  3.776e+01  -0.386   0.6998
## age25-34         -1.477e+01  3.776e+01  -0.391   0.6956
## age35-44         -1.472e+01  3.776e+01  -0.390   0.6967
## age45-54         -1.464e+01  3.776e+01  -0.388   0.6983
## age55-64         -1.461e+01  3.776e+01  -0.387   0.6988
## age65-74         -1.466e+01  3.776e+01  -0.388   0.6978
## dtir1             1.249e-03  7.397e-04   1.689   0.0912 .
## loan_amount      -4.296e-07  8.138e-08  -5.279 1.30e-07 ***
## property_value    1.067e-06  5.300e-08  20.139  < 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: 151921  on 135115  degrees of freedom
## Residual deviance: 147146  on 135099  degrees of freedom
## AIC: 147180
##
## Number of Fisher Scoring iterations: 12
```

```
car::vif(model)
```

```
##                      GVIF Df GVIF^(1/(2*Df))
## loan_type        1.334219  2        1.074748
## rate_of_interest 1.187856  1        1.089888
## term             1.207085  1        1.098674
## income           1.763263  1        1.327879
## Credit_Score     1.000116  1        1.000058
## age              1.210455  7        1.013736
## dtir1            1.145929  1        1.070481
## loan_amount      3.562284  1        1.887401
## property_value   2.804807  1        1.674756
```

```
# Visualization
#install.packages("corrplot")
library(corrplot)
```

```
## corrplot 0.92 loaded
```

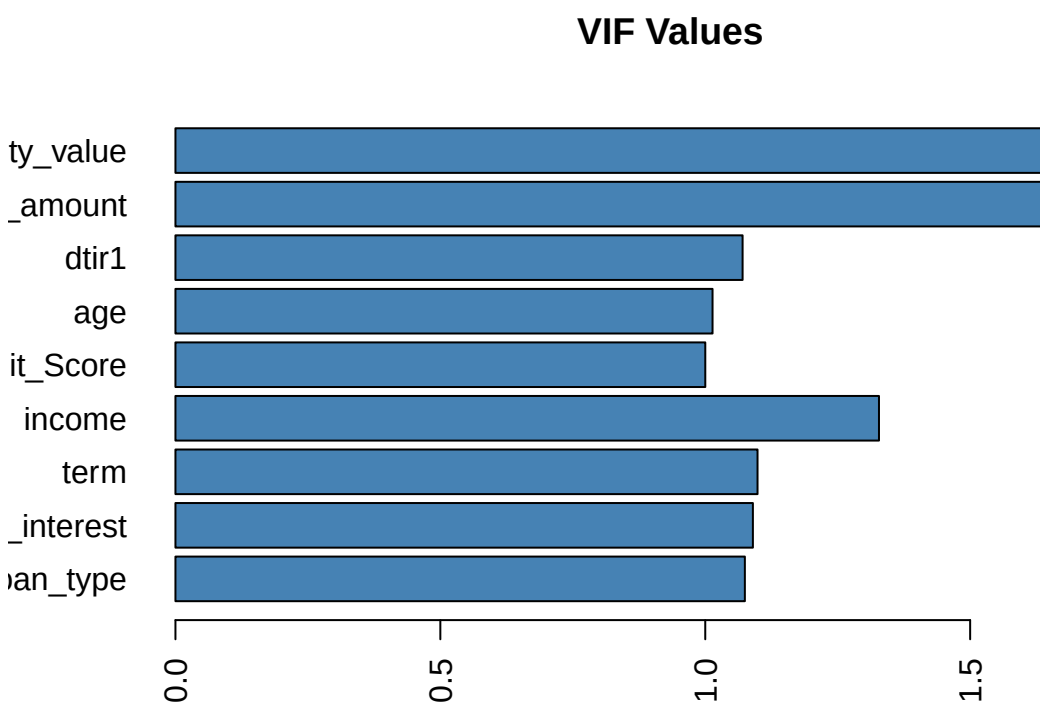
```
vif_values <- vif(model)           #create vector of VIF values
#barplot(vif_values, main = "VIF Values", horiz = TRUE, col = "steelblue") #create horizontal bar chart
#abline(v = 5, lwd = 3, lty = 2)    #add vertical line at 5 as after 5 there is severe correlation

# Basing on the results, it looks like our model does not have any severe multicollinearity, as the bas
```

```
d <- data.frame(vif_values)
d
```

```
##                      GVIF Df GVIF..1..2.Df..
## loan_type        1.334219  2        1.074748
## rate_of_interest 1.187856  1        1.089888
## term             1.207085  1        1.098674
## income           1.763263  1        1.327879
```

```
## Credit_Score     1.000116  1        1.000058
## age              1.210455  7        1.013736
## dtir1            1.145929  1        1.070481
## loan_amount      3.562284  1        1.887401
## property_value   2.804807  1        1.674756
```

```
barplot(d[,3], main = "VIF Values", col = "steelblue", names=row.names(d),las=2,horiz=TRUE) #create
```




```
#install.packages("bestglm")
library(bestglm)
vifx(vif_values)
```

```
##            GVIF             Df GVIF^(1/(2*Df))
##      135.031024       2.168063      141.667454
```

##andrew's code

Step 1. Splitting the dataset into train and test dataset

```
sample <- sample.int(n = nrow(imputed_data), size = floor(.8*nrow(imputed_data)), replace = F) #80% of

train <- imputed_data[sample, ]
test <- imputed_data[-sample, ]
summary(train)
```

```
##      Status       loan_type      loan_amount     rate_of_interest
##  Min.   :0.0000   type1:80342   Min.   : 16500   Min.   :0.000
##  1st Qu.:0.0000   type2:16482   1st Qu.:186500   1st Qu.:3.750
##  Median :0.0000   type3:11268   Median :286500   Median :4.045
##  Mean   :0.2505                 Mean   :306386   Mean   :4.048
##  3rd Qu.:1.0000                 3rd Qu.:406500   3rd Qu.:4.250
##  Max.   :1.0000                 Max.   :766500   Max.   :8.000
##
```

```
##       term       property_value       income       Credit_Score
##  Min.   : 96.0   Min.   :  8000   Min.   :    0   Min.   :500.0
##  1st Qu.:360.0   1st Qu.:278000   1st Qu.: 3660   1st Qu.:600.0
##  Median :360.0   Median :428000   Median : 5640   Median :700.0
##  Mean   :335.6   Mean   :435197   Mean   : 5945   Mean   :699.9
##  3rd Qu.:360.0   3rd Qu.:538000   3rd Qu.: 7500   3rd Qu.:800.0
##  Max.   :360.0   Max.   :998000   Max.   :14940   Max.   :900.0
##
##       age            dtir1
##  45-54  :24432   Min.   : 5.00
##  35-44  :23959   1st Qu.:34.00
##  55-64  :23321   Median :37.73
##  65-74  :15287   Mean   :38.19
##  25-34  :14647   3rd Qu.:44.00
##  >74    : 5251   Max.   :61.00
##  (Other): 1195
```

##create a model with all the variables

```
model_intial = glm(Status ~ loan_type+loan_amount+rate_of_interest+term+property_value+income+Credit_Sc
summary(model_intial)
```

```
##
## Call:
## glm(formula = Status ~ loan_type + loan_amount + rate_of_interest +
##     term + property_value + income + Credit_Score + age + dtir1,
##     family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.31928  -0.78862  -0.68627   0.00154   2.36731
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       1.392e+01  4.184e+01   0.333   0.7394
## loan_typetype2    5.337e-01  2.030e-02  26.290  < 2e-16 ***
## loan_typetype3    1.405e-01  2.539e-02   5.536 3.10e-08 ***
## loan_amount      -4.369e-07  9.119e-08  -4.791 1.66e-06 ***
## rate_of_interest  1.936e-02  1.644e-02   1.178   0.2388
## term             -7.061e-04  1.357e-04  -5.202 1.97e-07 ***
## property_value    1.042e-06  5.929e-08  17.569  < 2e-16 ***
## income           -1.274e-04  3.486e-06 -36.548  < 2e-16 ***
## Credit_Score      5.483e-05  6.164e-05   0.890   0.3737
## age<25           -1.458e+01  4.184e+01  -0.349   0.7274
## age>74           -1.455e+01  4.184e+01  -0.348   0.7281
## age25-34         -1.480e+01  4.184e+01  -0.354   0.7236
## age35-44         -1.472e+01  4.184e+01  -0.352   0.7250
## age45-54         -1.464e+01  4.184e+01  -0.350   0.7264
## age55-64         -1.460e+01  4.184e+01  -0.349   0.7271
## age65-74         -1.466e+01  4.184e+01  -0.350   0.7261
## dtir1             1.844e-03  8.264e-04   2.231   0.0257 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 121686  on 108091  degrees of freedom
## Residual deviance: 117881  on 108075  degrees of freedom
## AIC: 117915
##
## Number of Fisher Scoring iterations: 12
```

#model with varaibles of significance from previous model

```
options(scipen=9)
model_sig = glm(Status ~ loan_type+loan_amount+rate_of_interest+term+property_value+income+dtir1, family
summary(model_sig)
```

```
##
## Call:
## glm(formula = Status ~ loan_type + loan_amount + rate_of_interest +
##     term + property_value + income + dtir1, family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3465  -0.7895  -0.6907   1.0925   2.3382
##
## Coefficients:
##                       Estimate     Std. Error z value          Pr(>|z|)
## (Intercept)      -0.74854465022  0.08216380876  -9.110           < 2e-16 ***
## loan_typetype2    0.55497616027  0.02010981771  27.597           < 2e-16 ***
## loan_typetype3    0.16857219008  0.02500974231   6.740 0.000000000015810 ***
## loan_amount      -0.00000065305  0.00000008783  -7.436 0.000000000000104 ***
## rate_of_interest  0.03616588856  0.01634947662   2.212            0.0270 *
## term             -0.00078758884  0.00013491327  -5.838 0.000000005291304 ***
## property_value    0.00000112510  0.00000005779  19.468           < 2e-16 ***
## income           -0.00012438175  0.00000344278 -36.128           < 2e-16 ***
## dtir1             0.00200728290  0.00082436879   2.435            0.0149 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 121686  on 108091  degrees of freedom
## Residual deviance: 118413  on 108083  degrees of freedom
## AIC: 118431
##
## Number of Fisher Scoring iterations: 4
```

#lets find variables that are interacting

```
model_findi = glm(Status ~ (loan_type+loan_amount+rate_of_interest+term+property_value+income)^2, family
summary(model_findi)
```

```
##
## Call:
## glm(formula = Status ~ (loan_type + loan_amount + rate_of_interest +
##     term + property_value + income)^2, family = binomial(link = "logit"),
##     data = train)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.5186  -0.7765  -0.5683   0.1593   3.6233
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -2.102e+01  6.383e-01 -32.939  < 2e-16 ***
## loan_typetype2                  -5.299e+00  2.898e-01 -18.284  < 2e-16 ***
## loan_typetype3                  -1.139e+01  3.962e-01 -28.755  < 2e-16 ***
## loan_amount                     -1.215e-05  1.188e-06 -10.229  < 2e-16 ***
## rate_of_interest                 4.165e+00  1.507e-01  27.642  < 2e-16 ***
## term                             8.785e-02  1.823e-03  48.205  < 2e-16 ***
## property_value                  -2.417e-06  7.565e-07  -3.195   0.0014 **
## income                           3.541e-04  4.258e-05   8.317  < 2e-16 ***
## loan_typetype2:loan_amount      -4.430e-06  3.193e-07 -13.875  < 2e-16 ***
## loan_typetype3:loan_amount      -1.480e-06  3.284e-07  -4.508 6.54e-06 ***
## loan_typetype2:rate_of_interest  2.045e+00  5.559e-02  36.791  < 2e-16 ***
## loan_typetype3:rate_of_interest  3.218e+00  6.862e-02  46.899  < 2e-16 ***
## loan_typetype2:term             -7.024e-03  5.409e-04 -12.986  < 2e-16 ***
## loan_typetype3:term             -4.811e-03  7.565e-04  -6.359 2.03e-10 ***
## loan_typetype2:property_value    6.280e-06  2.540e-07  24.721  < 2e-16 ***
## loan_typetype3:property_value    5.941e-06  2.812e-07  21.130  < 2e-16 ***
## loan_typetype2:income           -1.999e-04  1.061e-05 -18.839  < 2e-16 ***
## loan_typetype3:income           -2.536e-04  1.311e-05 -19.343  < 2e-16 ***
## loan_amount:rate_of_interest     2.065e-06  2.517e-07   8.202 2.36e-16 ***
## loan_amount:term                 6.981e-10  1.856e-09   0.376   0.7067
## loan_amount:property_value      -5.313e-12  3.801e-13 -13.979  < 2e-16 ***
## loan_amount:income               1.110e-09  2.906e-11  38.192  < 2e-16 ***
## rate_of_interest:term           -1.988e-02  4.270e-04 -46.553  < 2e-16 ***
## rate_of_interest:property_value  4.140e-06  1.689e-07  24.513  < 2e-16 ***
## rate_of_interest:income         -8.004e-05  9.333e-06  -8.576  < 2e-16 ***
## term:property_value             -2.268e-08  1.147e-09 -19.778  < 2e-16 ***
## term:income                     -3.902e-08  6.081e-08  -0.642   0.5211
## property_value:income           -9.838e-10  2.825e-11 -34.820  < 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: 121686  on 108091  degrees of freedom
## Residual deviance: 106475  on 108064  degrees of freedom
## AIC: 106531
##
## Number of Fisher Scoring iterations: 5
```

```
anova(model_findi, test='Chisq')
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Status
##
## Terms added sequentially (first to last)
##
```

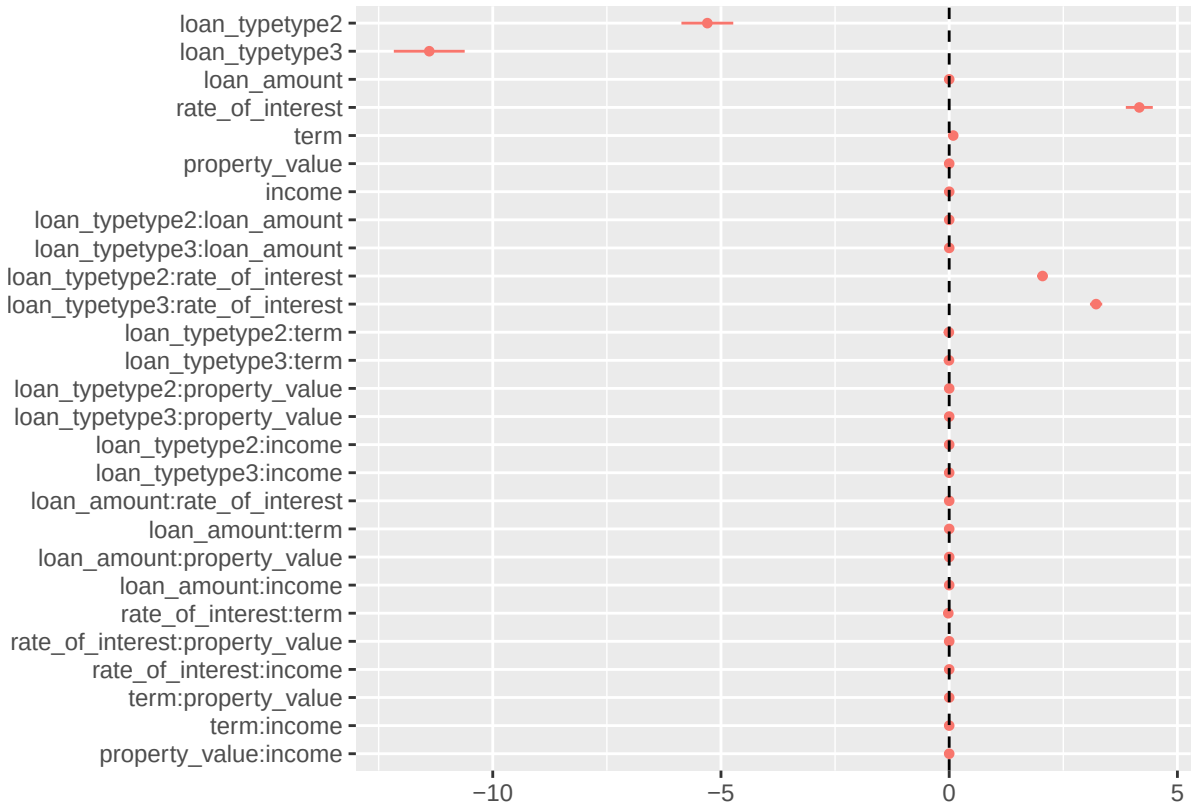
```
##
##                                  Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                             108091     121686
## loan_type                         2   938.82    108089     120748 < 2.2e-16 ***
## loan_amount                       1   367.48    108088     120380 < 2.2e-16 ***
## rate_of_interest                  1     0.50    108087     120380    0.4790
## term                              1     0.34    108086     120379    0.5586
## property_value                    1   366.32    108085     120013 < 2.2e-16 ***
## income                            1  1594.09    108084     118419 < 2.2e-16 ***
## loan_type:loan_amount             2    67.92    108082     118351 1.787e-15 ***
## loan_type:rate_of_interest        2  1856.38    108080     116495 < 2.2e-16 ***
## loan_type:term                    2    51.28    108078     116443 7.312e-12 ***
## loan_type:property_value          2  1867.69    108076     114576 < 2.2e-16 ***
## loan_type:income                  2   244.10    108074     114331 < 2.2e-16 ***
## loan_amount:rate_of_interest      1  1055.10    108073     113276 < 2.2e-16 ***
## loan_amount:term                  1   109.51    108072     113167 < 2.2e-16 ***
## loan_amount:property_value        1  1229.76    108071     111937 < 2.2e-16 ***
## loan_amount:income                1   621.24    108070     111316 < 2.2e-16 ***
## rate_of_interest:term             1  2713.34    108069     108602 < 2.2e-16 ***
## rate_of_interest:property_value   1   498.94    108068     108104 < 2.2e-16 ***
## rate_of_interest:income           1    75.86    108067     108028 < 2.2e-16 ***
## term:property_value               1   218.34    108066     107809 < 2.2e-16 ***
## term:income                       1     0.24    108065     107809    0.6264
## property_value:income             1  1334.47    108064     106475 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on the ANOVA test conducted above, the following interactions were deemed significant based on their deviances (>10), as well as p-values(<0.05):

loan_type:rate_of_interest 2 1874.67 107831 115980 < 2.2e-16 ***loan_type:property_value 2 1859.09 107827 114066 < 2.2e-16*** rate_of_interest:term 1 2745.61 107820 108094 < 2.2e-16 ***

For our model testing we will use loan_type:income 2 244.10 108074 114331 < 2.2e-16 ***

```
#install.packages("dotwhisker")
library(dotwhisker)
dwplot(model_findi,ci_method="wald")+geom_vline(xintercept=0,lty=2)
```




```
summary(train)
```

```
##      Status       loan_type      loan_amount     rate_of_interest
##  Min.   :0.0000   type1:80342   Min.   : 16500   Min.   :0.000
##  1st Qu.:0.0000   type2:16482   1st Qu.:186500   1st Qu.:3.750
##  Median :0.0000   type3:11268   Median :286500   Median :4.045
##  Mean   :0.2505                 Mean   :306386   Mean   :4.048
##  3rd Qu.:1.0000                 3rd Qu.:406500   3rd Qu.:4.250
##  Max.   :1.0000                 Max.   :766500   Max.   :8.000
##
##       term       property_value       income       Credit_Score
##  Min.   : 96.0   Min.   :  8000   Min.   :    0   Min.   :500.0
##  1st Qu.:360.0   1st Qu.:278000   1st Qu.: 3660   1st Qu.:600.0
##  Median :360.0   Median :428000   Median : 5640   Median :700.0
##  Mean   :335.6   Mean   :435197   Mean   : 5945   Mean   :699.9
##  3rd Qu.:360.0   3rd Qu.:538000   3rd Qu.: 7500   3rd Qu.:800.0
##  Max.   :360.0   Max.   :998000   Max.   :14940   Max.   :900.0
##
##       age            dtir1
##  45-54  :24432   Min.   : 5.00
##  35-44  :23959   1st Qu.:34.00
##  55-64  :23321   Median :37.73
##  65-74  :15287   Mean   :38.19
##  25-34  :14647   3rd Qu.:44.00
##  >74    : 5251   Max.   :61.00
##  (Other): 1195
```

```
unique(train$loan_type)
```

```
## [1] type1 type2 type3
## Levels: type1 type2 type3
```

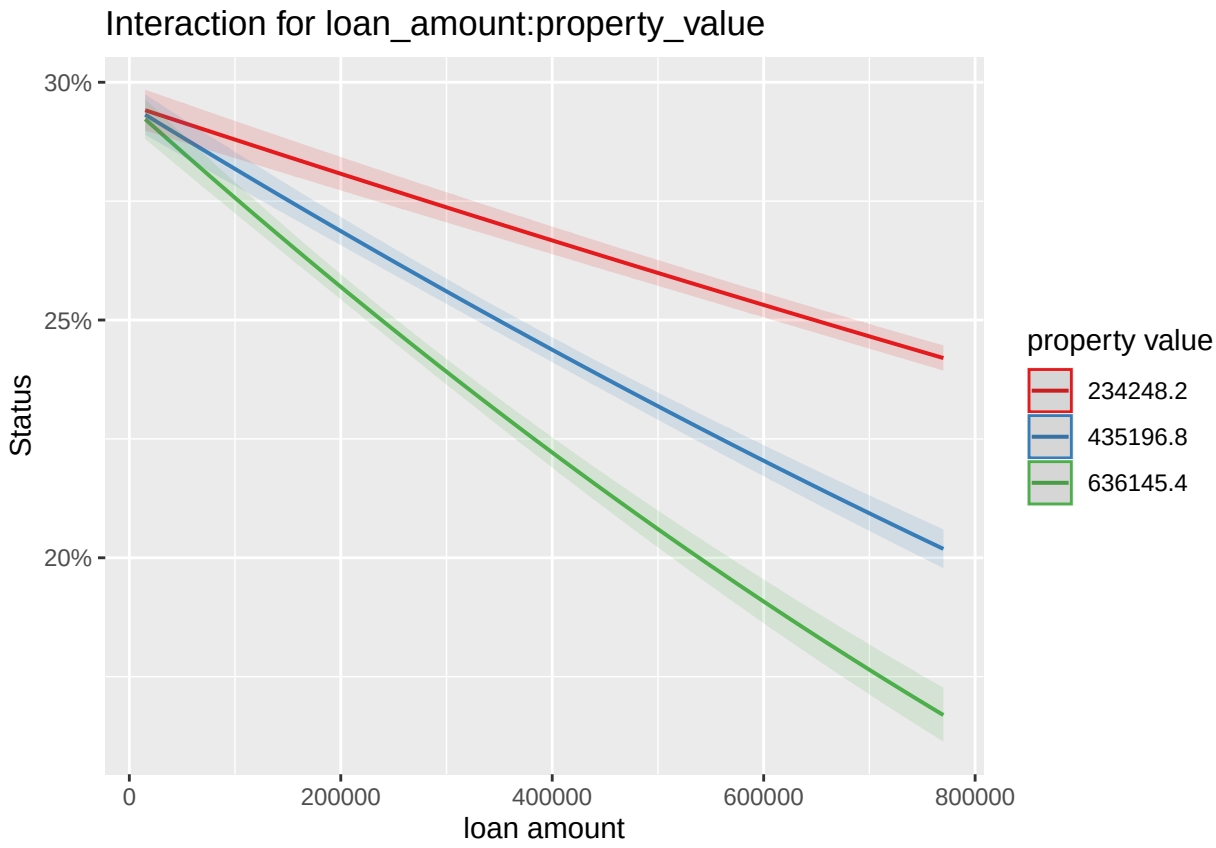
```
as.numeric(as.factor(unique(train$loan_type)))
```

```
## [1] 1 2 3
```

#intercation plots

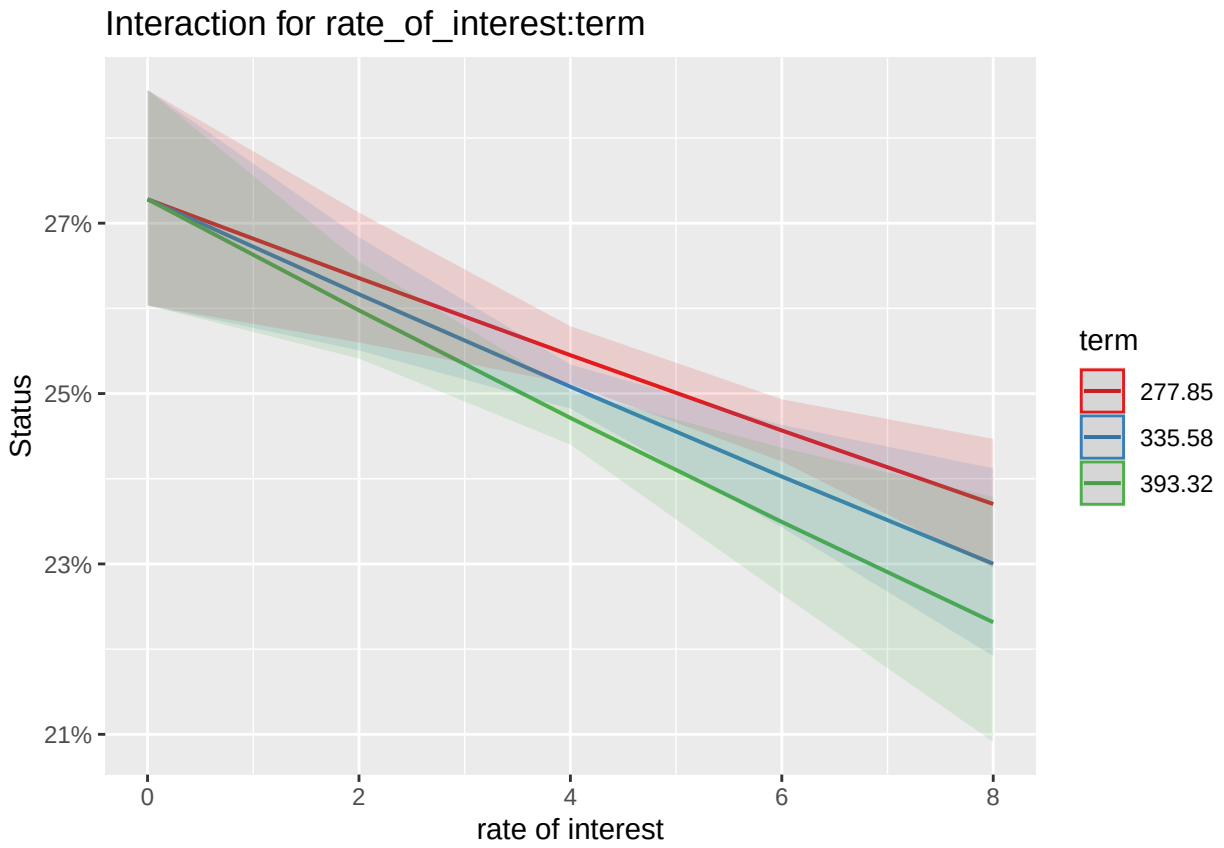
```
library(sjPlot)
fit <- glm(Status ~ loan_amount:property_value, data = train,family = binomial)
plot_model(fit, type = "int",mdrt.values = "meansd",title="Interaction for loan_amount:property_value")
```

```
## Data were 'prettified'. Consider using `terms="loan_amount [all]"` to
##   get smooth plots.
```



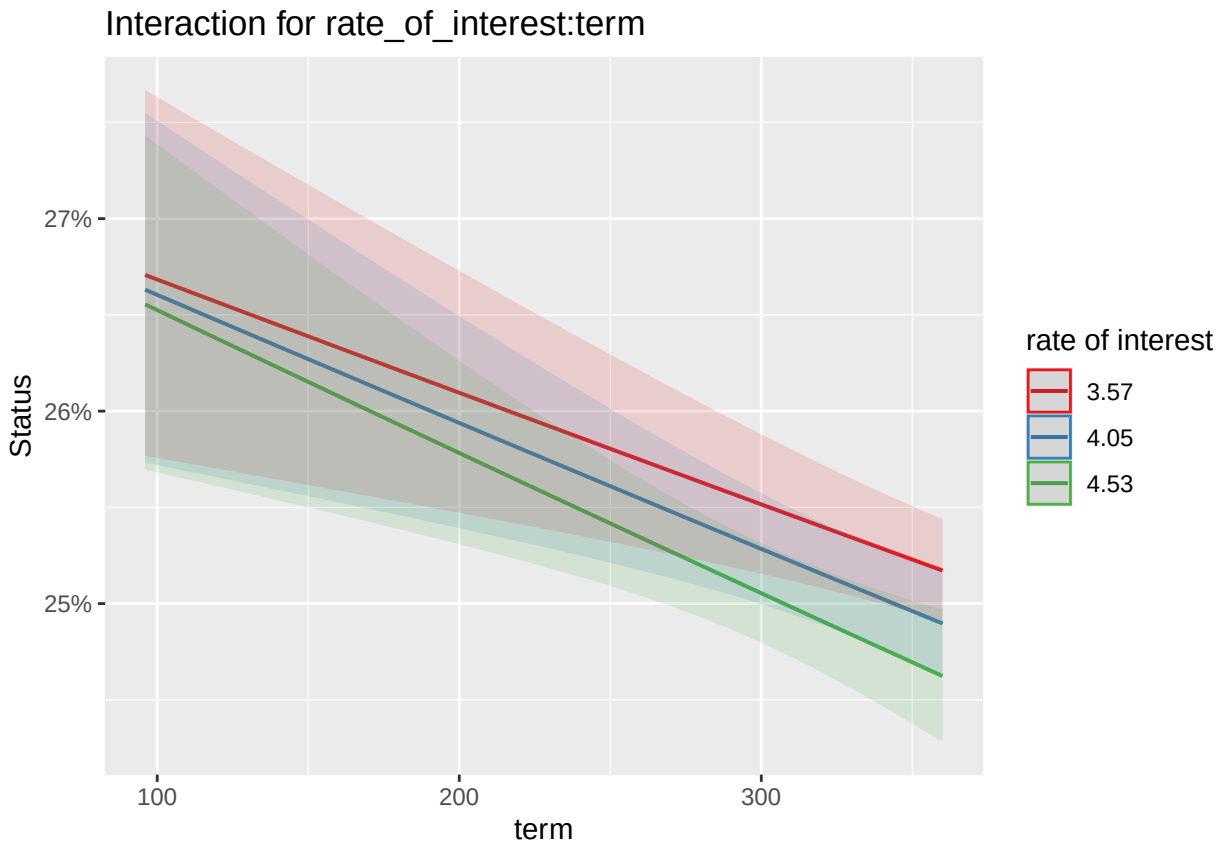

```
fit <- glm(Status ~ rate_of_interest:term , data = train,family = binomial)
plot_model(fit, type = "int",mdrt.values = "meansd",title="Interaction for rate_of_interest:term")
```

```
## Data were 'prettified'. Consider using `terms="rate_of_interest [all]"`
##   to get smooth plots.
```




```
fit <- glm(Status ~ term:rate_of_interest , data = train,family = binomial)
plot_model(fit, type = "int",mdrt.values = "meansd",title="Interaction for rate_of_interest:term ")
```

```
## Data were 'prettified'. Consider using `terms="term [all]"` to get
##   smooth plots.
```




```
summary(fit)
```

```
##
## Call:
## glm(formula = Status ~ term:rate_of_interest, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7982  -0.7610  -0.7554   1.6304   1.6875
##
## Coefficients:
##                           Estimate  Std. Error z value Pr(>|z|)
## (Intercept)           -0.98043351  0.03252205 -30.147  < 2e-16 ***
## term:rate_of_interest -0.00008484  0.00002336  -3.633 0.000281 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 121686  on 108091  degrees of freedom
## Residual deviance: 121673  on 108090  degrees of freedom
## AIC: 121677
##
## Number of Fisher Scoring iterations: 4
```

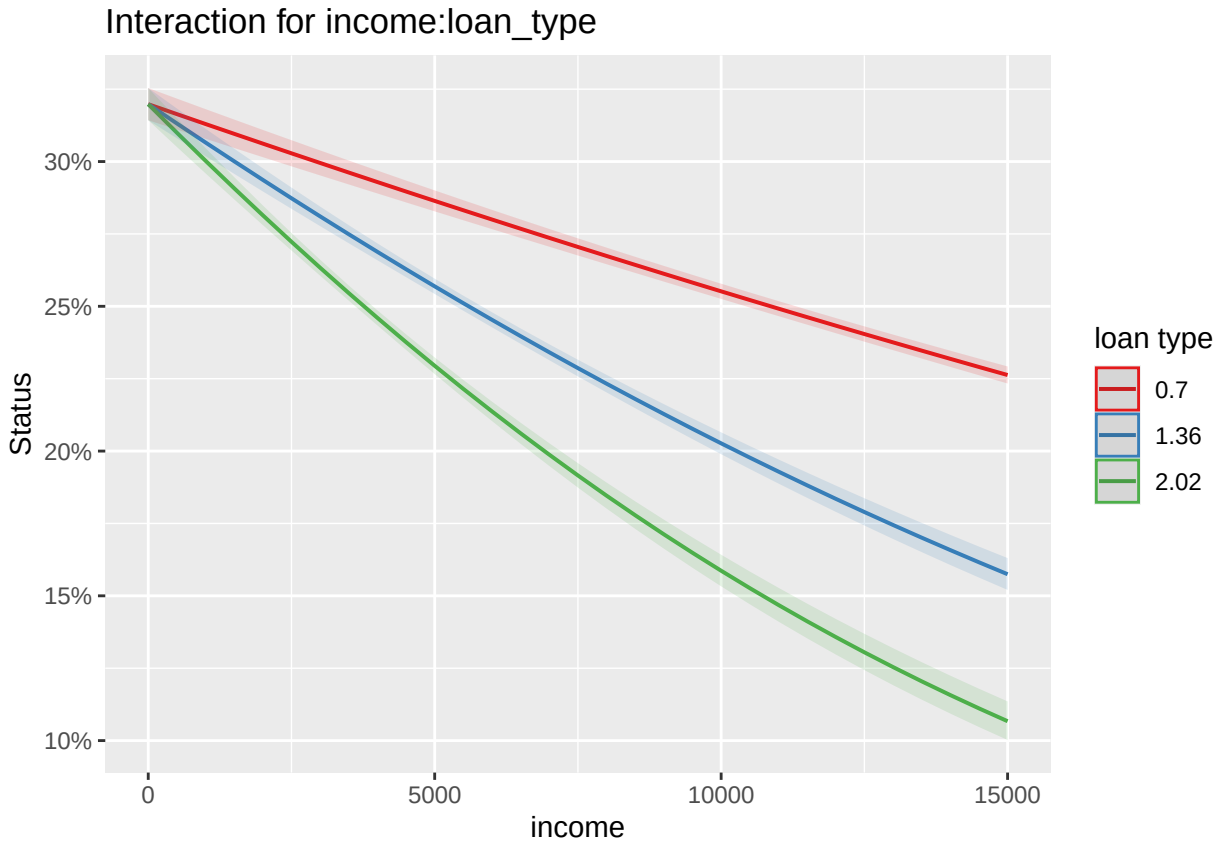
```
x <- train
x$loan_type <- as.numeric(as.factor(x$loan_type))
```

```r
fit <- glm(Status ~ income:loan_type , data = x,family = "binomial"(link="logit"))
summary(fit)
```

```
##
## Call:
## glm(formula = Status ~ income:loan_type, family = binomial(link = "logit"),
##     data = x)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8779  -0.7947  -0.7423   1.5100   2.3840
##
## Coefficients:
##                       Estimate   Std. Error z value Pr(>|z|)
## (Intercept)      -0.754627974  0.013050488  -57.82   <2e-16 ***
## income:loan_type -0.000045225  0.000001516  -29.83   <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: 121686  on 108091  degrees of freedom
## Residual deviance: 120717  on 108090  degrees of freedom
## AIC: 120721
##
## Number of Fisher Scoring iterations: 4
```

```r
plot_model(fit, type = "int",mdrt.values = "meansd",title="Interaction for income:loan_type ")
```

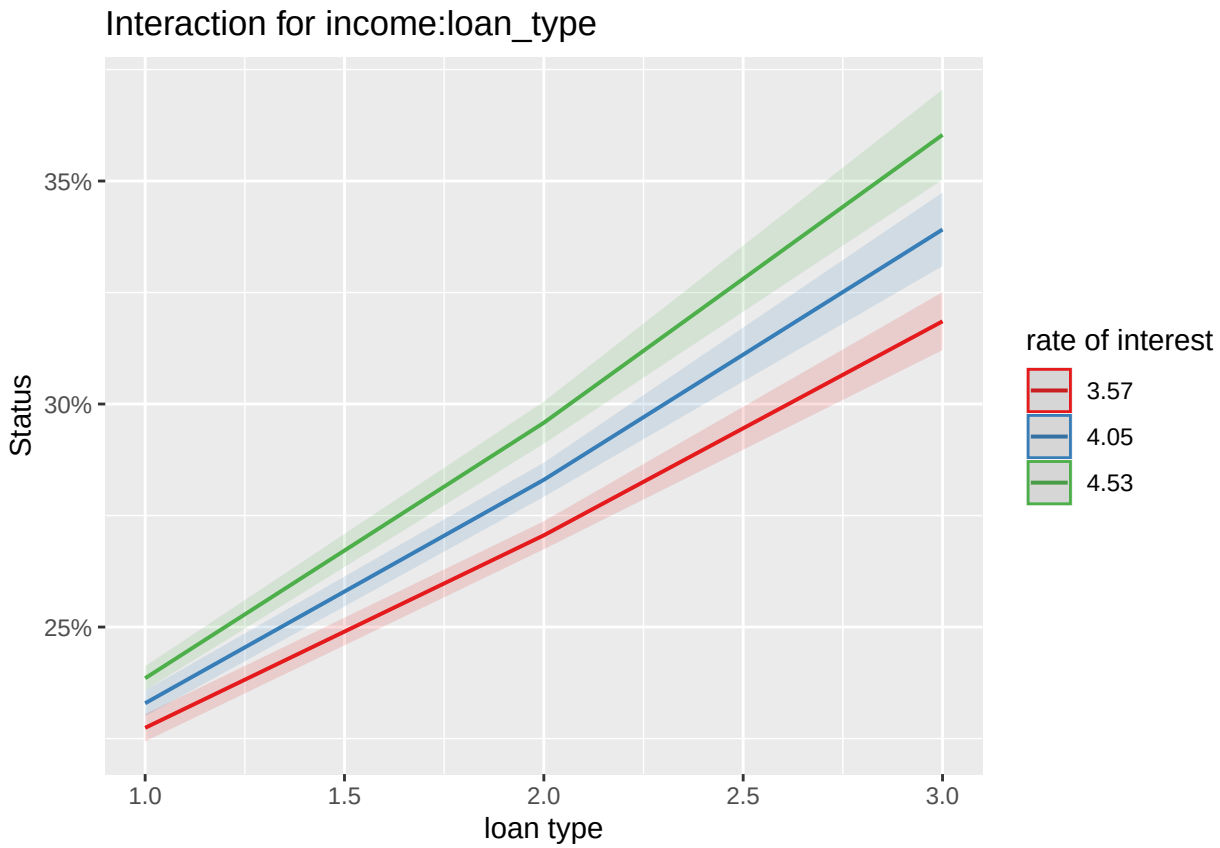
```
## Data were 'prettified'. Consider using `terms="income [all]"` to get
##   smooth plots.
```




```
fit <- glm(Status ~ loan_type:rate_of_interest , data = x,family = "binomial"(link="logit"))
summary(fit)
```

```
##
## Call:
## glm(formula = Status ~ loan_type:rate_of_interest, family = binomial(link = "logit"),
##     data = x)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0381  -0.7428  -0.7270   1.4710   1.7340
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -1.454328   0.016575  -87.74   <2e-16 ***
## loan_type:rate_of_interest  0.064787   0.002668   24.29   <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: 121686  on 108091  degrees of freedom
## Residual deviance: 121113  on 108090  degrees of freedom
## AIC: 121117
##
## Number of Fisher Scoring iterations: 4
```

```
plot_model(fit, type = "int",mdrt.values = "meansd",title="Interaction for income:loan_type ")
```

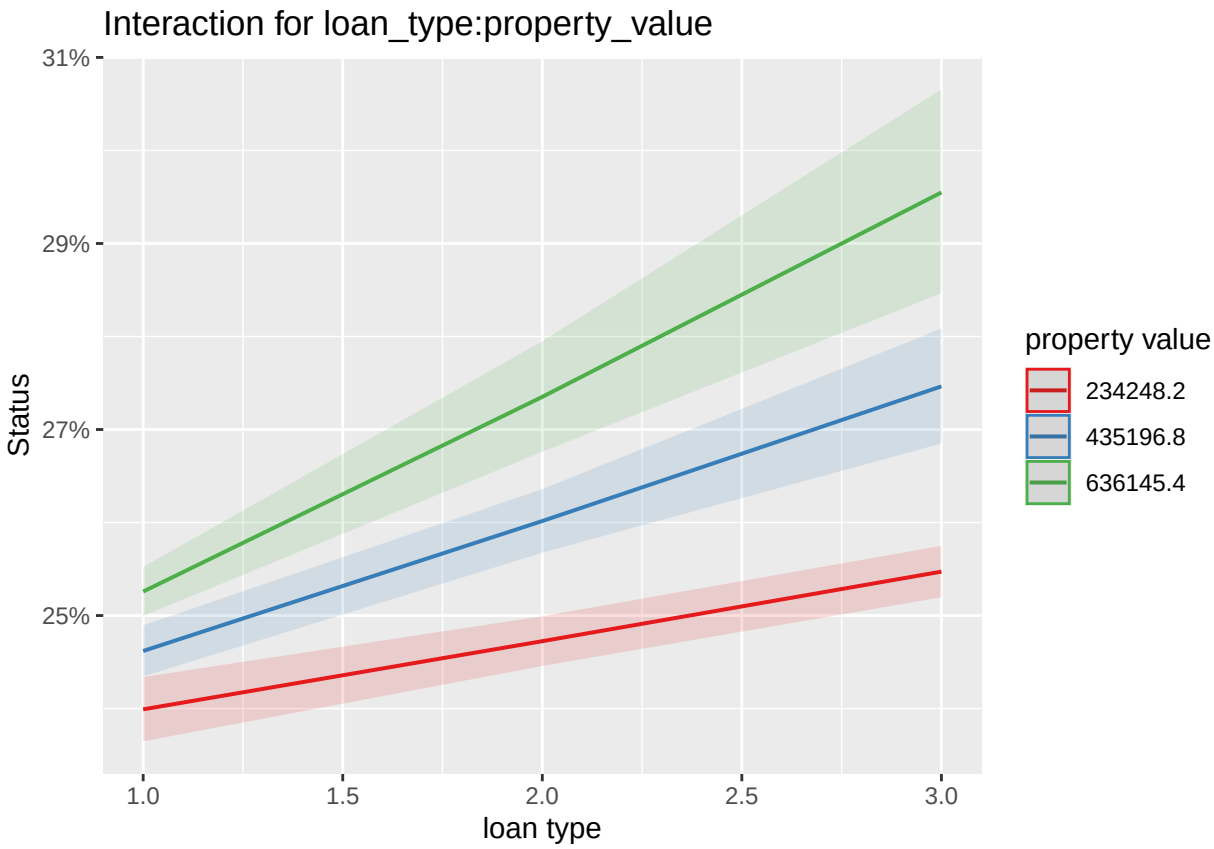



```
fit <- glm(Status ~ loan_type:property_value , data = x,family = "binomial"(link="logit"))
summary(fit)
```

```
##
## Call:
## glm(formula = Status ~ loan_type:property_value, family = binomial(link = "logit"),
##     data = x)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9037  -0.7615  -0.7486   1.5519   1.7063
##
## Coefficients:
##                                Estimate     Std. Error z value Pr(>|z|)
## (Intercept)              -1.19293418709  0.01322973275 -90.171   <2e-16 ***
## loan_type:property_value  0.00000016981  0.00000001946   8.727   <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: 121686  on 108091  degrees of freedom
## Residual deviance: 121611  on 108090  degrees of freedom
## AIC: 121615
##
```

```
## Number of Fisher Scoring iterations: 4
```

```
plot_model(fit, type = "int",mdrt.values = "meansd",title="Interaction for loan_type:property_value ")
```



#intercation model

```
model_interactive = glm(Status ~ loan_type+loan_amount+rate_of_interest+term+property_value+income+incor
summary(model_interactive)
```

```
##
## Call:
## glm(formula = Status ~ loan_type + loan_amount + rate_of_interest +
##     term + property_value + income + income:loan_type, family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4497  -0.7676  -0.6888   0.9433   2.8248
##
## Coefficients:
##                            Estimate     Std. Error z value     Pr(>|z|)
## (Intercept)          -0.72098586859  0.07696298335  -9.368      < 2e-16 ***
## loan_typetype2        1.08018450825  0.04121781262  26.207      < 2e-16 ***
## loan_typetype3        1.16319143525  0.05743483452  20.252      < 2e-16 ***
## loan_amount          -0.00000052224  0.00000008608  -6.067 0.0000000013 ***
## rate_of_interest     -0.00101612165  0.01655540604  -0.061        0.951
## term                 -0.00063363909  0.00013518998  -4.687 0.0000027720 ***
## property_value        0.00000101037  0.00000005802  17.414      < 2e-16 ***
```

```
## income                -0.00009607961  0.00000351462 -27.337      < 2e-16 ***
## loan_typetype2:income -0.00010965840  0.00000778633 -14.083      < 2e-16 ***
## loan_typetype3:income -0.00019544578  0.00001048013 -18.649      < 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: 121686  on 108091  degrees of freedom
## Residual deviance: 117903  on 108082  degrees of freedom
## AIC: 117923
##
## Number of Fisher Scoring iterations: 4
```

Step 2. Comparing the interaction model and non-interaction model

Residual Deviance, F-test, which one is more fit? Non-Interactive Model: Residual deviance: 130653 on 118919 degrees of freedom Interactive Model: Residual deviance: 129660 on 118917 degrees of freedom AIC: 129678

```
pchisq((130653 - 129660), (118919-118917), lower.tail=FALSE)
```

```
## [1] 2.359336e-216
```

```
# the p-value is approximately 0. This suggests that the reduced model is appropriate
```

#lrest

```
library(lmtest)
lrtest(model_sig,model_interactive)
```

```
## Likelihood ratio test
##
## Model 1: Status ~ loan_type + loan_amount + rate_of_interest + term +
##     property_value + income + dtir1
## Model 2: Status ~ loan_type + loan_amount + rate_of_interest + term +
##     property_value + income + income:loan_type
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1   9 -59206
## 2  10 -58952  1 509.42  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Step 4. Model Evaluation Using ROC Curves

Classification report

#model withh alll variables: model_intial #model withh all significnat variabless: model_sig #model with interactive variables:model_interactive

```
#test the model with all significant variables against test data
test$non_interactive_fitted <- predict(model_sig, test)

test$non_interactive_prob <- exp(test$non_interactive_fitted)/(1+exp(test$non_interactive_fitted))

# The Classification report: Sensitivity, Specificity and Accuracy
prob_cut_off = sum(test$Status)/nrow(test)
```

```
test$non_interactive_predict <- as.numeric(test$non_interactive_prob > prob_cut_off)

# 2. The interactive model: Only keeping interaction terms
test$interactive_fitted <- predict(model_interactive, test)

test$interactive_prob <- exp(test$interactive_fitted)/(1+exp(test$interactive_fitted))

test$interactive_predict <- as.numeric(test$interactive_prob > prob_cut_off)
```

```
library(caret)

#test the model with all significant variables against test data
threshold=prob_cut_off
predicted_values<-ifelse(predict(model_sig, test)>threshold,1,0)
actual_values<-test$Status
conf_matrix<-table(predicted_values,actual_values)
conf_matrix
```

```
##                 actual_values
## predicted_values     0     1
##                0 20339  6681
##                1     1     3
```

```
cat("\nsensitivity:",sensitivity(conf_matrix))
```

```
##
## sensitivity: 0.9999508
```

```
#posPredValue(conf_matrix)
cat("\nposPredValue:",posPredValue(conf_matrix))
```

```
##
## posPredValue: 0.7527387
```

```
#specificity(conf_matrix)
cat("\nspecificity:",specificity(conf_matrix))
```

```
##
## specificity: 0.000448833
```

```
cat("\nnegPredValue:",negPredValue(conf_matrix))
```

```
##
## negPredValue: 0.75
```

```
cat("\nMisclassifcation :",(conf_matrix[2,1]+conf_matrix[1,2])/sum(conf_matrix))
```

```
##
## Misclassifcation : 0.2472617
```

```
#test the interaction model  against test data
threshold=prob_cut_off
predicted_values<-ifelse(predict(model_interactive, test)>threshold,1,0)
actual_values<-test$Status
conf_matrix<-table(predicted_values,actual_values)
conf_matrix
```

```
##                 actual_values
```

```
## predicted_values     0     1
##                0 20338  6471
##                1     2   213
```

```
cat("\nsensitivity:",sensitivity(conf_matrix))
```

```
##
## sensitivity: 0.9999017
```

```
#posPredValue(conf_matrix)
cat("\nposPredValue:",posPredValue(conf_matrix))
```

```
##
## posPredValue: 0.7586258
```

```
#specificity(conf_matrix)
cat("\nspecificity:",specificity(conf_matrix))
```

```
##
## specificity: 0.03186715
```

```
cat("\nnegPredValue:",negPredValue(conf_matrix))
```

```
##
## negPredValue: 0.9906977
```

```
cat("\nMisclassifcation :",(conf_matrix[2,1]+conf_matrix[1,2])/sum(conf_matrix))
```

```
##
## Misclassifcation : 0.2395278
```

```
#just evaluating our full model
threshold=0.5
predicted_values<-ifelse(predict(model_sig,type="response")>threshold,1,0)
actual_values<-model_sig$y
conf_matrix<-table(predicted_values,actual_values)
conf_matrix
```

```
##                 actual_values
## predicted_values     0     1
##                0 80999 26513
##                1    16   564
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following object is masked from 'package:Metrics':
##
##     auc
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

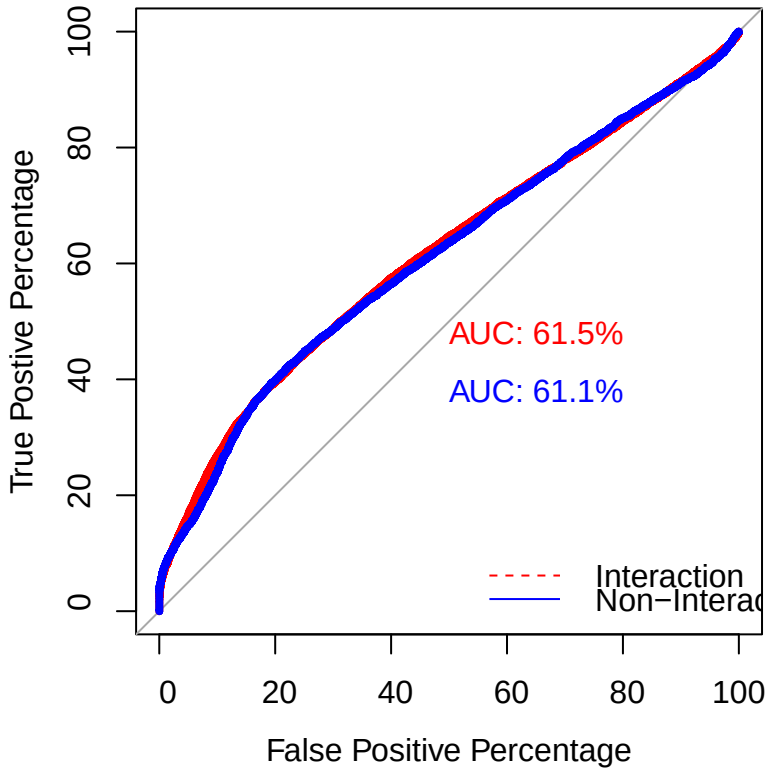
```
par(pty = "s")

roc(test$Status, test$non_interactive_prob, plot=TRUE, legacy.axes=TRUE, percent=TRUE, xlab="False Posi
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
##
## Call:
## roc.default(response = test$Status, predictor = test$non_interactive_prob,     percent = TRUE, plot =
##
## Data: test$non_interactive_prob in 20340 controls (test$Status 0) < 6684 cases (test$Status 1).
## Area under the curve: 61.53%
```

```
plot.roc(test$Status, test$interactive_prob, percent=TRUE, col="blue", lwd=4, print.auc=TRUE, add=TRUE,
```

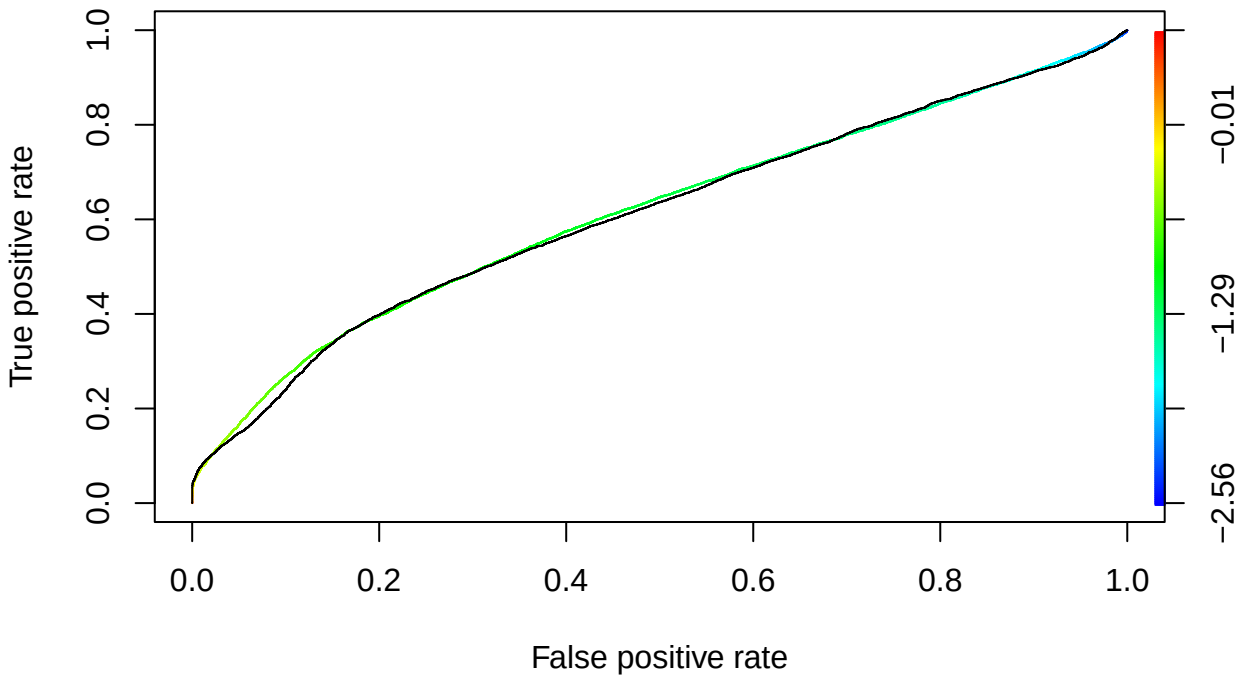
```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
legend("bottomright", legend=c("Interaction", "Non-Interaction"),
       col=c("red", "blue"), lwd=1, y.intersp = 0.5,
       lty = c(2,1),cex=1, bty ='n',pt.cex = 2)
```




```
#install.packages("ROCR")

library(ROCR)
p1 <- predict(model_sig, test)
pred <- prediction(p1, test$Status )
perf <- performance( pred, "tpr", "fpr" )

p2 <- predict(model_interactive, test)
pred2 <- prediction(p2, test$Status)
perf2 <- performance(pred2, "tpr", "fpr")

plot( perf, colorize = TRUE)
```

```
plot(perf2, add = TRUE, colorize = FALSE)
```




```
predicted_prob<-predict(model_interactive, test)
roccurve <- roc(test$Status, predicted_prob)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
predicted_prob1<-predict(model_sig, test)
roccurve1 <- roc(test$Status, predicted_prob1)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

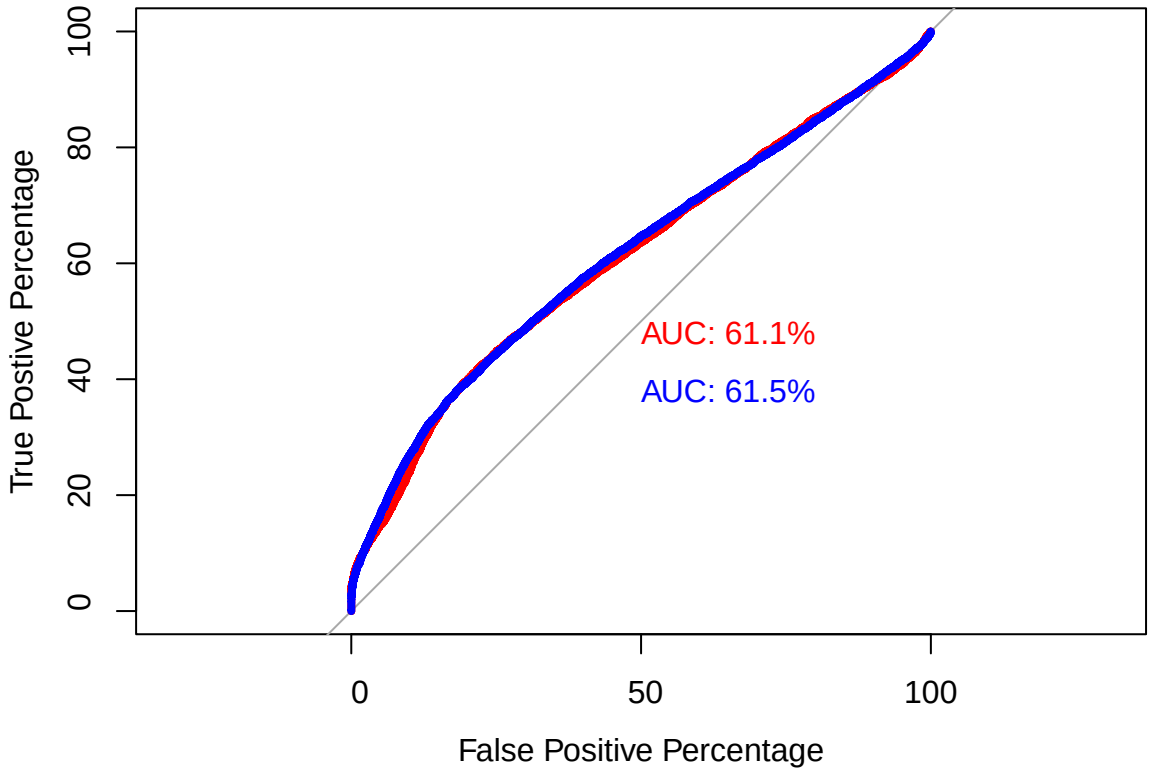
```
roc(test$Status, predicted_prob, plot=TRUE, legacy.axes=TRUE, percent=TRUE, xlab="False Positive Percen
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
##
## Call:
## roc.default(response = test$Status, predictor = predicted_prob,     percent = TRUE, plot = TRUE, lega
##
## Data: predicted_prob in 20340 controls (test$Status 0) < 6684 cases (test$Status 1).
## Area under the curve: 61.12%
```

```
plot.roc(test$Status, predicted_prob1, percent=TRUE, col="blue", lwd=4, print.auc=TRUE, add=TRUE, print
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```




```
#plot(roccurve1)
```

The Hosmer-Lemshow test

```
library(generalhoslem)
```

```
## Loading required package: reshape
```

```
##
## Attaching package: 'reshape'
```

```
## The following objects are masked from 'package:reshape2':
##
##     colsplit, melt, recast
```

```
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
## The following object is masked from 'package:dplyr':
##
##     rename
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, smiths
```

```
logitgof(test$Status, test$non_interactive_prob)
```

```
##
##  Hosmer and Lemeshow test (binary model)
##
```

```
## data:  test$Status, test$non_interactive_prob
## X-squared = 487.94, df = 8, p-value < 2.2e-16
```

```
logitgof(test$Status, test$interactive_prob)
```

```
##
##  Hosmer and Lemeshow test (binary model)
##
## data:  test$Status, test$interactive_prob
## X-squared = 360.11, df = 8, p-value < 2.2e-16
```

```
# There is evidence the model is fitting badly

library(ResourceSelection)
```

```
## ResourceSelection 0.3-5   2019-07-22
```

```
hoslem.test(test$Status, test$non_interactive_prob, g=10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  test$Status, test$non_interactive_prob
## X-squared = 487.94, df = 8, p-value < 2.2e-16
```

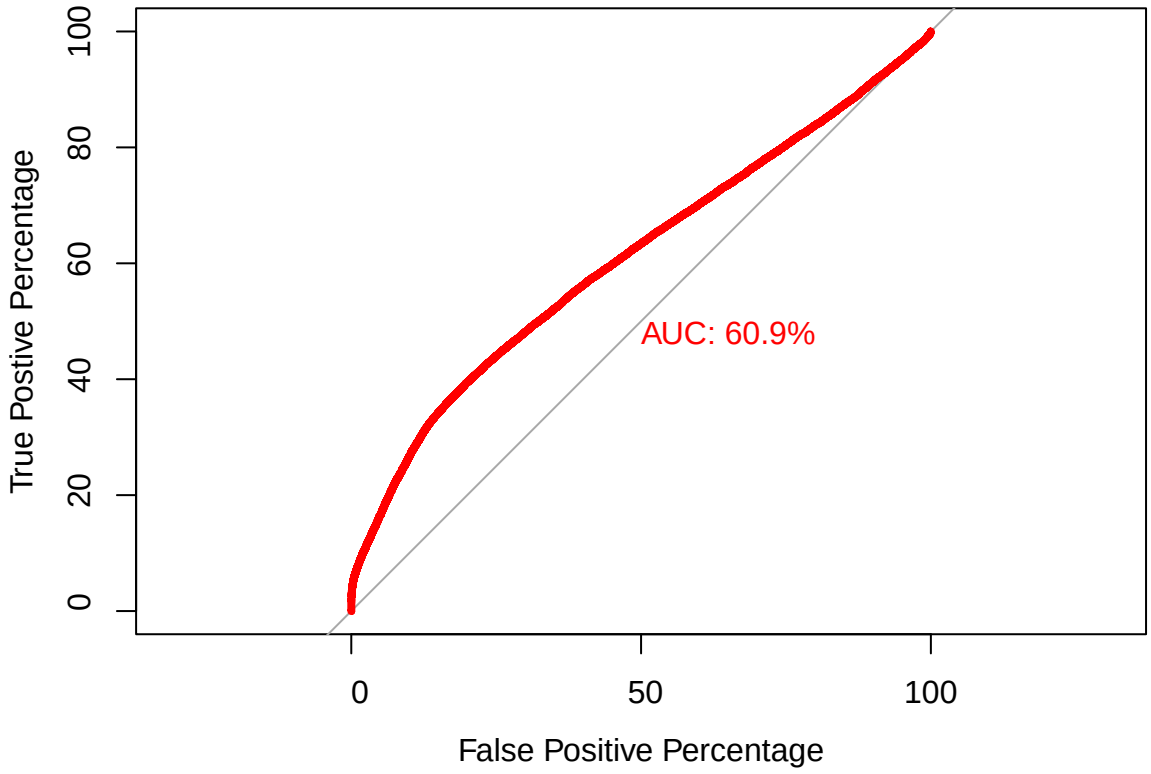
```
hoslem.test(test$Status, test$interactive_prob, g=10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  test$Status, test$interactive_prob
## X-squared = 360.11, df = 8, p-value < 2.2e-16
```

```
# 1. The non-interactive model

train$non_interactive_fitted <- predict(model_sig, train)

train$non_interactive_prob <- exp(train$non_interactive_fitted)/(1+exp(train$non_interactive_fitted))

roc(train$Status, train$non_interactive_prob, plot=TRUE, legacy.axes=TRUE, percent=TRUE, xlab="False Pos
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```




```
##
## Call:
## roc.default(response = train$Status, predictor = train$non_interactive_prob,     percent = TRUE, plo
##
## Data: train$non_interactive_prob in 81015 controls (train$Status 0) < 27077 cases (train$Status 1).
## Area under the curve: 60.94%
```

```
logitgof(train$Status, train$non_interactive_prob)
```

```
##
##  Hosmer and Lemeshow test (binary model)
##
## data:  train$Status, train$non_interactive_prob
## X-squared = 2137, df = 8, p-value < 2.2e-16
```