Credit Risk Model

Ganbayar Ulziibayar(10385860), Chuge Huang(9544224), Drashti Kanubhai Nayak(10599568)

1 November 2021

Introduction

In order to comply with the strict regulatory requirements, this report is to rebuild the statistic model regarding credit risk (loan default) to predict loan default based on information known at the time of loan application. To finalize this task, the preprocess of dataset has been done initially like cleaning data, dealing with missing data and covariates modeling. Then, providing with proper visualizations and insights interpretation, the significant processes will be demonstrated step by step, such as data exploration, selecting from a space of candidate models, link function comparation with final selection, testing assumptions of the model, validating the assumptions of the model, model performance evaluation and uncertainty of variables' effects. Ultimately. all the insights will be concluded to address the major problems showing below:

1. How does this model perform compared to the one you used previously? How can it be expected to perform on new loan applications? ->
2. What are the important variables in this model and how do they compare to variables that are traditionally important for predicting credit risk in the banking sector? -> Traditionally the demographic and social-demographic factors like age, home_ownership, income, dti, credit score, location, interest_rate, any past deliquency history, economical histories, account behaviour, listed securities etc. In this project we received 37 variables. From which we extracted 15 variables among which one is dependent variable called repay_fail which shows the state of default into 0's and 1's in numeric format. Among other variables we considered "loan amount" used to compare with annual income and installment amount to compare whether it affect dependent variable or not. The model showcase if the term length is 60 compared to 36 then default significance increases vastly. Interest showed 100% variable importance; if high becomes demotivating factors in paying loans. Employment Length shows stabiltiy in income which can result in on-time payment. The home ownership variable showed extreme importance when borrower is living on a rental property which increases it chances of becoming defaulter. People with mortgage showed mild significance. Annual income one of the important variables and hase been observed in the model as well. Verification_status shows validity on the annual income variable but hence it's included. Purpose defined most categories falls in personal loans so haven't further considered in model building. Dti is seen as an essential significance as it describes determine your mortgage eligibility and the likelihood you will repay a loan. Deliquency showcase borrower's time punctuality in payments. Traditionally two payment doesn't affect much however on 3rd miss payment it can result in severe decrease in credit risk. Henc ewe will factorise if below 2 the applicant is safe otherwise risky. Further this economic background exmined by inquries made in last 6 months and pub record which show derogatory information and considered negation by lenders. it indicates risk and hurts your ability to qualify for credit or other services because they reflect financial obligations that were not paid as agreed. They are further manipulated into categories inquires if more than six have chances of 8 times to default and hence that's our range for inquires. Pub record must be 0 otherwise risky. revolving utilisation rare show current revolving credit divided by total available. It is suggested to always keep it under 30%. To access credit risk initially during the time of application so many variable that shows information after loan application and non-relevant variables like total_acc, issu_d, installment amount, funded amount, member id and id have been discarded.

Load credit risk data into R

```
#### Read in data
dfo <- read_csv("final_data.csv",show_col_types = FALSE)
df <- copy(dfo)
```

Data frame overviews

The customer loan data contains 38480 observations with 37 variables. To be specific, loan id and member_id are identifiers. Also, We have information on funded amount committed to loan in the time and information on term length and interest rate. Besides, lenders' information like employment length, home_ownership status, annual income, purpose, address and debt-to-income ratio is shown in dataset. In addition, important variables like loan status and repay_fail, which represents whether the borrower has defaulted or not. In order to select valuable data for analysis, we will exclude variables that are not useful and extract variables that show application and behavioral information about borrower. Also, a distinctive feature of predicting default model needs independent categorical variables, which results in simplified model perform as a score card. Hence, all independent variables discrete or continuous will be converted into suitable categorical or dummy variable (e.g. Verification Status). Next, the NA and invalid values will be addressed in data cleaning and data pre-processing sections.

Data Pre-processing and Cleaning

Cleaning the data

We consider 16 variables as valuable items, which are home ownership records, state address, verification status, purpose, term of loan agreement, employment length, loan amount, interest rate, credit records of delinquency in last 2 years, delinquency records of last 6 month, the number of open accounts, public credit records, total number of accounts, annual income, dti ratios and repay fail records. Then, for better future analysis, we deal with NA values and convert original data into more proper data types. The process has been shown below:

Missing data handling

```
sum(is.na(df1))
```

```
## [1] 65
```

```
df1 <- na.omit(df1)
df2 <- df1[!(df1$loan_amnt == "0"), ]
df2$term <- str_replace(df2$term, "months", "")
```

The rows having NA values in the data frame have been removed, since it only has 65 observations that contain the NA values. 2. The observations of 0 values in loan amount, interest rate and annual income have been removed, since they make no sense for analysis.

Data Manipulation and Assumptions: variable-to-variable

Variable(emp_length)

```
df2$emp_length <- str_replace(df2$emp_length, "years", "")
df2$emp_length <- str_replace(df2$emp_length, "year", "")
df2$emp_length <- str_replace(df2$emp_length, "n/a", "0")
df2$emp_length <- str_replace(df2$emp_length, "\\+", "")
```

```
df2$emp_length <- str_replace(df2$emp_length, "<", "")
df2$emp_length <- str_replace(df2$emp_length, " 1", "1")
```

Since employment length has n/a string values, then values with special characters like <1 year and 10+years which must be converted into integer and where contains n/a string will be converted into 1 employment experience.

Variable(home_ownership)

```
df2$home_ownership <- str_replace(df2$home_ownership, "NONE", "OTHER")
```

home_ownership transfer none data rows into other which mostly suggest borrowers who lives with parents or lives in some other person's property.

Variable(verification_status)

```
df2$verification_status <- str_replace(
  df2$verification_status,
  "Source Verified", "Verified"
)
```

verification_status convert source verified into verified. we will assume the application must be assessed before verification which helps prediction based on other important demographic factors.

Variable(loan_status)

```
# convert loan_status into either paid or charged off
df2$loan_status <- str_replace(
  df2$loan_status,
  "Does not meet the credit policy. Status:Fully Paid", "Fully Paid"
)
df2$loan_status <- str_replace(
  df2$loan_status,
  "Current", "Fully Paid"
)
df2$loan_status <- str_replace(
  df2$loan_status,
  "In Grace Period", "Fully Paid"
)
df2$loan_status <- str_replace(
  df2$loan_status,
  "Does not meet the credit policy. Status:Charged Off", "Charged Off"
)
df2$loan_status <- str_replace(
  df2$loan_status,
  "Default", "Charged Off"
)
df2$loan_status[df2$loan_status == "Late (16-30 days)"] <- "Charged Off"
df2$loan_status[df2$loan_status == "Late (31-120 days)"] <- "Charged Off"

# Based on the analysis. Most of the loans are personal loans.
# Hence, credit risk modelled based on personal loan data.
# However, we will once check the significance of purpose as it's important factor in credit risk.
# which purpose resulted in higher default will be analysed in EDA phase.
```

```
# Let's keep the application where status of previous loans is not verified.
df3 <- df2[!names(df2) %in% c("verification_status")]
df3$loan_status[df2$loan_status == "Fully Paid"] <- "0"
df3$loan_status[df2$loan_status == "Charged Off"] <- "1"
```

Variable(dti)

dti does not need any manipulation. pupose variable useful to see frequency of which type of loan in personal loans is more in defaulting.

Variable(delinq_2yrs)

```
df3 <- mutate(df3, delinq_2yrs = ifelse(delinq_2yrs <= 2, "0", "1"))
```

delinq_2yrs if less than two damage is minimal however after two missed payments credit score drop significantly and categorises borrower into risky.

Variable(inq_last_6mths)

```
df3 <- mutate(df3, inq_last_6mths = ifelse(inq_last_6mths < 6, "0", "1"))
```

Inquiry made <6 or more than 6 brings 8 time chances of bankruptcy(0 -> low inquires, 1-> higher inquiries brings risky borrowers).

Variable(pub_rec)

```
df3 <- mutate(df3, pub_rec = ifelse(pub_rec == 0, "0", "1"))
```

public record either 0 or else shows loans are not paid(0 -> safe and 1 -> risk).

Variable(revol_util)

```
df3$revol_util <- str_replace(df3$revol_util, "\\%", "")
df3 <- mutate(df3, revol_util = ifelse(revol_util < 30, "0", "1"))
```

revol_util must be below 30 or shows negation to lenders(if <30, 0 -> safe else 1 -> risk).

Converting variables

```
colnames(df3)
```

```
##  [1] "loan_amnt"      "term"           "int_rate"       "emp_length"
##  [5] "home_ownership" "annual_inc"     "loan_status"    "purpose"
##  [9] "dti"            "delinq_2yrs"    "inq_last_6mths" "pub_rec"
## [13] "revol_util"     "repay_fail"
```

```
# convert character values into numeric.
df3$loan_amnt <- as.numeric(df3$loan_amnt)
df3$int_rate <- as.numeric(df3$int_rate)
df3$annual_inc <- as.numeric(df3$annual_inc)
df3$dti <- as.numeric(df3$dti)

# convert categorical into dummy variables
df3$home_ownership <- factor(df3$home_ownership,
```

```
    levels = c("OWN", "MORTGAGE", "RENT", "OTHER"),
    labels = c("0", "1", "2", "3")
)
# Employment length factorise:
df3$emp_length <- as.factor(
  ifelse(df3$emp_length <= 2, "0-2",
    ifelse(df3$emp_length <= 5, "3-5",
      ifelse(df3$emp_length <= 8, "6-8", "10")
    )
  )
)

# Convert ordinal data

# convert numeric categorical variables into factors
df3$term <- as.factor(df3$term)
df3$loan_status <- as.factor(df3$loan_status)
df3$purpose <- as.factor(df3$purpose)
df3$delinq_2yrs <- as.factor(df3$delinq_2yrs)
df3$inq_last_6mths <- as.factor(df3$inq_last_6mths)
df3$pub_rec <- as.factor(df3$pub_rec)
df3$revol_util <- as.factor(df3$revol_util)
```

Convert character values into numeric such as loan amount, interest rate, annual income and dti ratios for continuous variables. Convert categorical into dummy variables such as home ownership and employment length.
Convert numeric categorical variables into factors such as loan term, loan status, delinquency in last 2 years, times of delinquency in last 6 months, public credit records and revol_util.

Exploratory data analysis

```
# Exploratory data set
df4 <- df3
```

Categorical variable analysis

```
### Single variable analysis
CrossTable(df4$home_ownership)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  38418
##
##
##           |         0 |         1 |         2 |         3 |
##           |-----------|-----------|-----------|-----------|
```

```
##            |      2951 |     17126 |     18214 |       127 |
##            |     0.077 |     0.446 |     0.474 |     0.003 |
##            |-----------|-----------|-----------|-----------|
##
##
##
##
```

```
CrossTable(df4$purpose)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  38418
##
##
##                     |                 car |         credit_card | debt_consolidation |         educationa
##                     |---------------------|---------------------|---------------------|-------------------
##                     |                1477 |                4970 |               17901 |                 38
##                     |               0.038 |               0.129 |               0.466 |               0.01
##                     |---------------------|---------------------|---------------------|-------------------
##
##
##                     |               house |     major_purchase |            medical |              movi
##                     |---------------------|---------------------|---------------------|-------------------
##                     |                 386 |                2071 |                 673 |                 5
##                     |               0.010 |               0.054 |               0.018 |               0.01
##                     |---------------------|---------------------|---------------------|-------------------
##
##
##                     |   renewable_energy |     small_business |           vacation |             weddi
##                     |---------------------|---------------------|---------------------|-------------------
##                     |                  91 |                1807 |                 359 |                 9
##                     |               0.002 |               0.047 |               0.009 |               0.01
##                     |---------------------|---------------------|---------------------|-------------------
##
##
##
##
```

```
CrossTable(df4$repay_fail)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
```

```
##
##
## Total Observations in Table:  38418
##
##
##           |         0 |         1 |
##           |-----------|-----------|
##           |     32610 |      5808 |
##           |     0.849 |     0.151 |
##           |-----------|-----------|
##
##
##
##
```

```
### With response variable
CrossTable(df4$repay_fail, df4$home_ownership,
  prop.r = TRUE, prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE
)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |-------------------------|
##
##
## Total Observations in Table:  38418
##
##
##                | df4$home_ownership
## df4$repay_fail |         0 |         1 |         2 |         3 | Row Total |
## ---------------|-----------|-----------|-----------|-----------|-----------|
##              0 |      2492 |     14684 |     15337 |        97 |     32610 |
##                |     0.076 |     0.450 |     0.470 |     0.003 |     0.849 |
## ---------------|-----------|-----------|-----------|-----------|-----------|
##              1 |       459 |      2442 |      2877 |        30 |      5808 |
##                |     0.079 |     0.420 |     0.495 |     0.005 |     0.151 |
## ---------------|-----------|-----------|-----------|-----------|-----------|
##   Column Total |      2951 |     17126 |     18214 |       127 |     38418 |
## ---------------|-----------|-----------|-----------|-----------|-----------|
##
##
```

```
CrossTable(df4$repay_fail, df4$purpose,
  prop.r = TRUE, prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE
)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
```

```
## |-------------------------|
##
##
## Total Observations in Table:  38418
##
##
##                 | df4$purpose
## df4$repay_fail |                 car |        credit_card | debt_consolidation |        educational |
## ---------------|--------------------|--------------------|--------------------|--------------------|
##              0 |                1315 |               4395 |              15126 |                301 |
##                |               0.040 |              0.135 |              0.464 |              0.009 |
## ---------------|--------------------|--------------------|--------------------|--------------------|
##              1 |                 162 |                575 |               2775 |                 82 |
##                |               0.028 |              0.099 |              0.478 |              0.014 |
## ---------------|--------------------|--------------------|--------------------|--------------------|
##   Column Total |                1477 |               4970 |              17901 |                383 |
## ---------------|--------------------|--------------------|--------------------|--------------------|
##
##
```

```
CrossTable(df4$repay_fail, df4$loan_status,
  prop.r = TRUE, prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE
)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |-------------------------|
##
##
## Total Observations in Table:  38418
##
##
##                 | df4$loan_status
## df4$repay_fail |         0 |         1 | Row Total |
## ---------------|-----------|-----------|-----------|
##              0 |     32610 |         0 |     32610 |
##                |     1.000 |     0.000 |     0.849 |
## ---------------|-----------|-----------|-----------|
##              1 |         0 |      5808 |      5808 |
##                |     0.000 |     1.000 |     0.151 |
## ---------------|-----------|-----------|-----------|
##   Column Total |     32610 |      5808 |     38418 |
## ---------------|-----------|-----------|-----------|
##
##
```

```
CrossTable(df4$repay_fail, df4$inq_last_6mths,
  prop.r = TRUE, prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE
)
```

```
##
##
```

```
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |-------------------------|
##
##
## Total Observations in Table:  38418
##
##
##                 | df4$inq_last_6mths
## df4$repay_fail |         0 |         1 | Row Total |
## ---------------|-----------|-----------|-----------|
##              0 |     32146 |       464 |     32610 |
##                |     0.986 |     0.014 |     0.849 |
## ---------------|-----------|-----------|-----------|
##              1 |      5568 |       240 |      5808 |
##                |     0.959 |     0.041 |     0.151 |
## ---------------|-----------|-----------|-----------|
##   Column Total |     37714 |       704 |     38418 |
## ---------------|-----------|-----------|-----------|
##
##
```

```
CrossTable(df4$repay_fail, df4$pub_rec,
  prop.r = TRUE, prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE
)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |-------------------------|
##
##
## Total Observations in Table:  38418
##
##
##                 | df4$pub_rec
## df4$repay_fail |         0 |         1 | Row Total |
## ---------------|-----------|-----------|-----------|
##              0 |     30975 |      1635 |     32610 |
##                |     0.950 |     0.050 |     0.849 |
## ---------------|-----------|-----------|-----------|
##              1 |      5306 |       502 |      5808 |
##                |     0.914 |     0.086 |     0.151 |
## ---------------|-----------|-----------|-----------|
##   Column Total |     36281 |      2137 |     38418 |
## ---------------|-----------|-----------|-----------|
##
##
```

```
CrossTable(df4$repay_fail, df4$revol_util,
  prop.r = TRUE, prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE
)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |-------------------------|
##
##
## Total Observations in Table:  38418
##
##
##                | df4$revol_util
## df4$repay_fail |         0 |         1 | Row Total |
## ---------------|-----------|-----------|-----------|
##              0 |      8144 |     24466 |     32610 |
##                |     0.250 |     0.750 |     0.849 |
## ---------------|-----------|-----------|-----------|
##              1 |      1071 |      4737 |      5808 |
##                |     0.184 |     0.816 |     0.151 |
## ---------------|-----------|-----------|-----------|
##   Column Total |      9215 |     29203 |     38418 |
## ---------------|-----------|-----------|-----------|
##
##
```

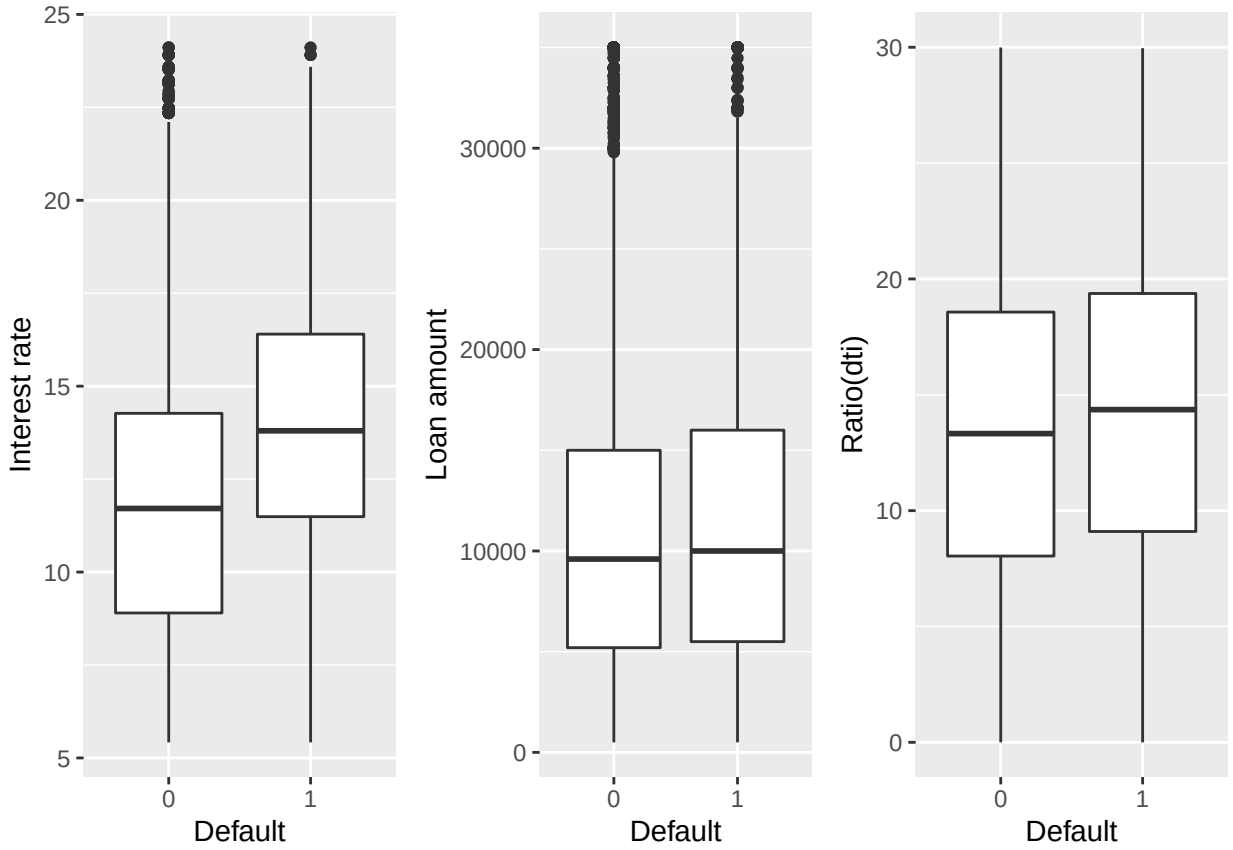
```
CrossTable(df4$repay_fail, df4$term,
  prop.r = TRUE, prop.c = FALSE, prop.t = FALSE, prop.chisq = FALSE
)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |-------------------------|
##
##
## Total Observations in Table:  38418
##
##
##                | df4$term
## df4$repay_fail |       36  |       60  | Row Total |
## ---------------|-----------|-----------|-----------|
##              0 |     25033 |      7577 |     32610 |
##                |     0.768 |     0.232 |     0.849 |
## ---------------|-----------|-----------|-----------|
##              1 |      3503 |      2305 |      5808 |
##                |     0.603 |     0.397 |     0.151 |
## ---------------|-----------|-----------|-----------|
```

```
##   Column Total |     28536 |      9882 |     38418 |
## ---------------|-----------|-----------|-----------|
##
##
```

Continuous variable analysis

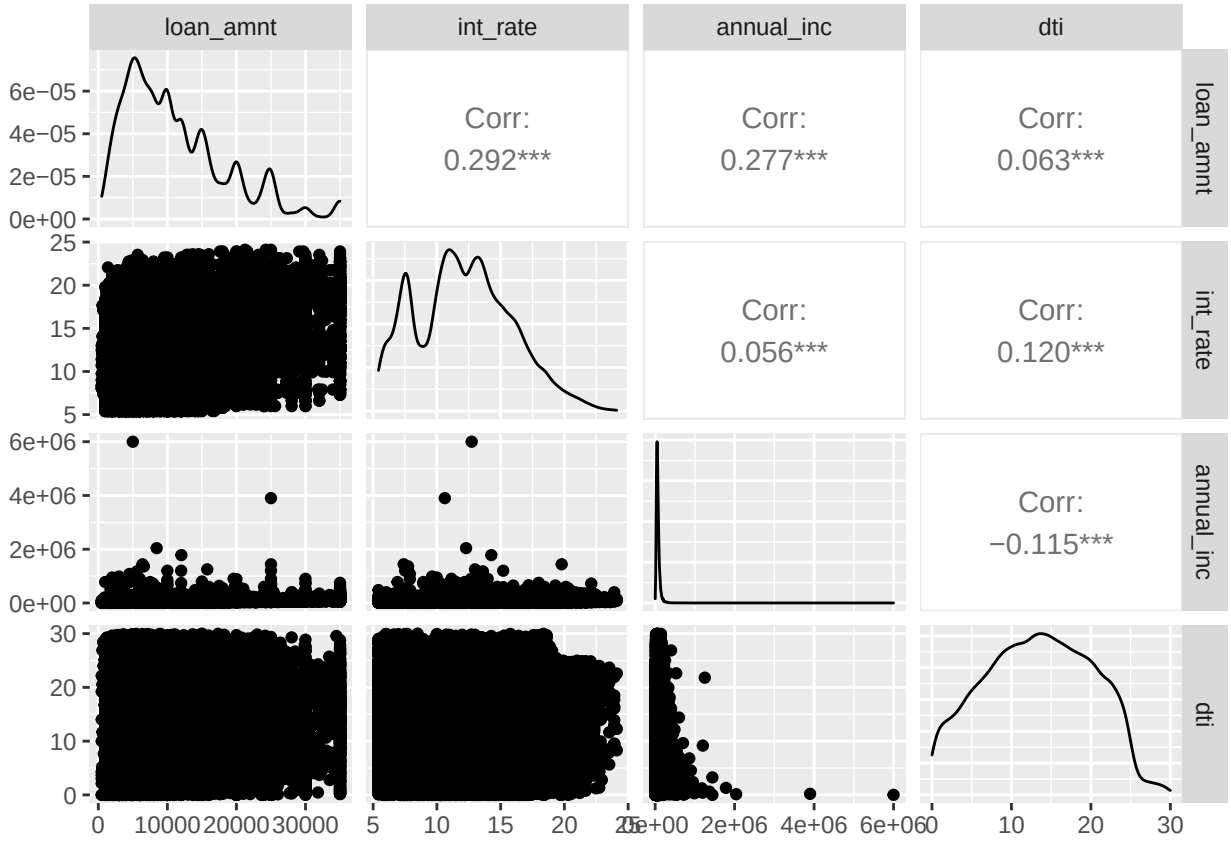
```
p1 <- ggplot(data = df4, aes(x = as.factor(repay_fail), y = int_rate)) +
    geom_boxplot() +
    labs(x = "Default", y = "Interest rate")
p2 <- ggplot(data = df4, aes(x = as.factor(repay_fail), y = loan_amnt)) +
    geom_boxplot() +
    labs(x = "Default", y = "Loan amount")
p3 <- ggplot(data = df4, aes(x = as.factor(repay_fail), y = dti)) +
    geom_boxplot() +
    labs(x = "Default", y = "Ratio(dti)")
ggarrange(p1, p2, p3, nrow = 1, ncol = 3)
```



For analysis, we create 3 boxplots to analyze the relationship regarding repay fail (default) compared with interest rate, loan amount and dti ratio respectively. For interest-default boxplot, it is easy to see that loans with higher interest rate are more likely resulting in fail to repay the instalments. Also, for loan amount and default boxplot, loan amount in default has higher volumes compared with loan amount in non-default. In addition, the mean value in default slightly higher than none-default, which means that higher loan amount has slightly more chance to result in fail to repay. For dti-default boxplot, it also has the trend that customers with higher dti ratio are likely to repay the instalments on time.

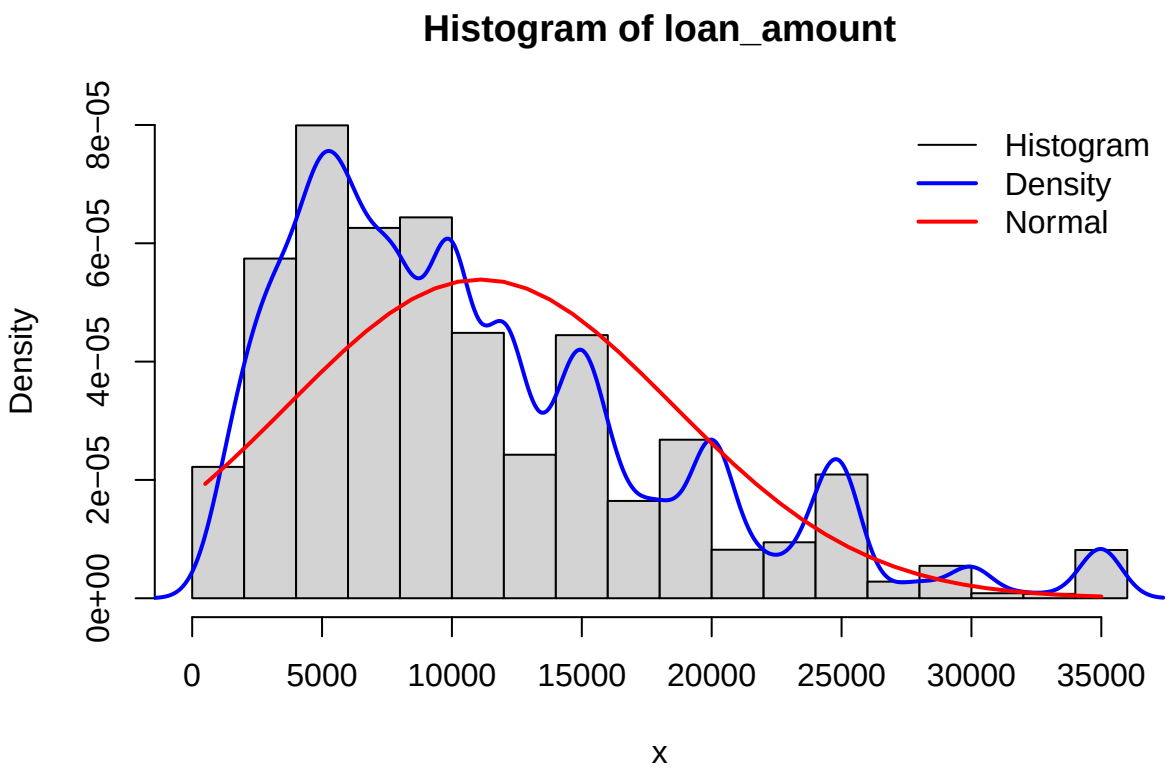
Correlation Matrix of continuous variables

```
ggpairs(df4[, c("loan_amnt", "int_rate", "annual_inc", "dti")])
```

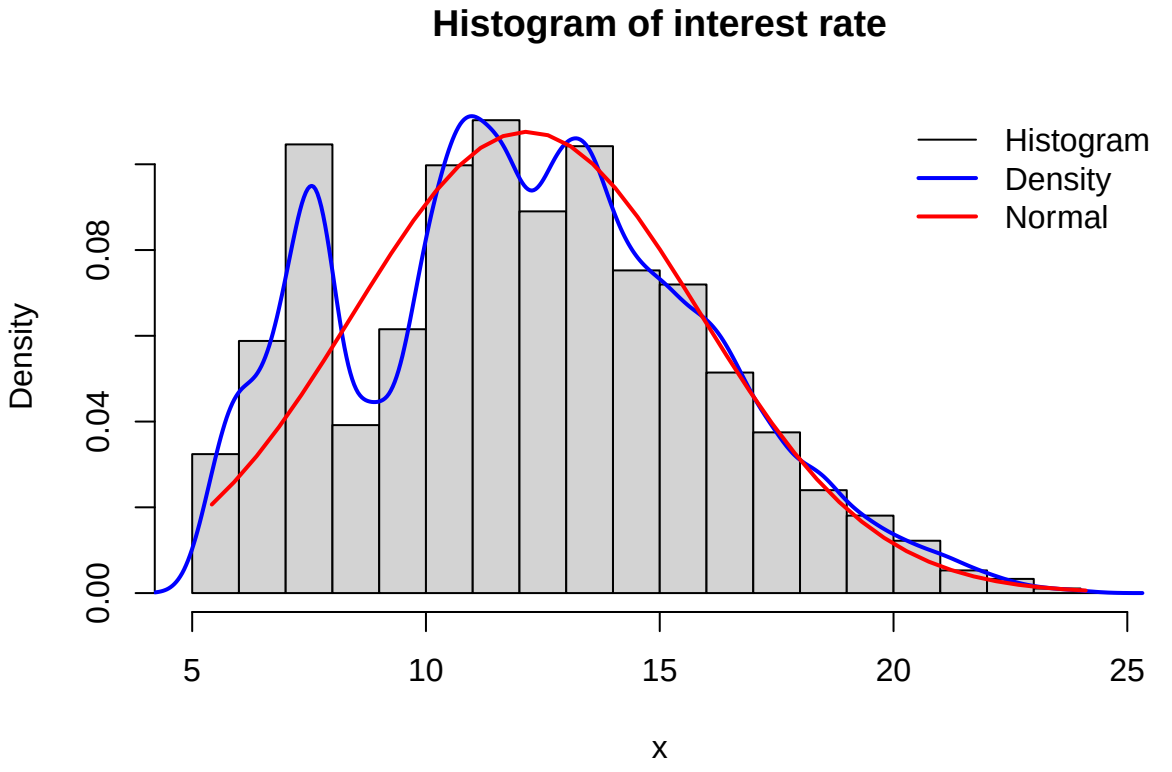


Histograms for continuous varibles

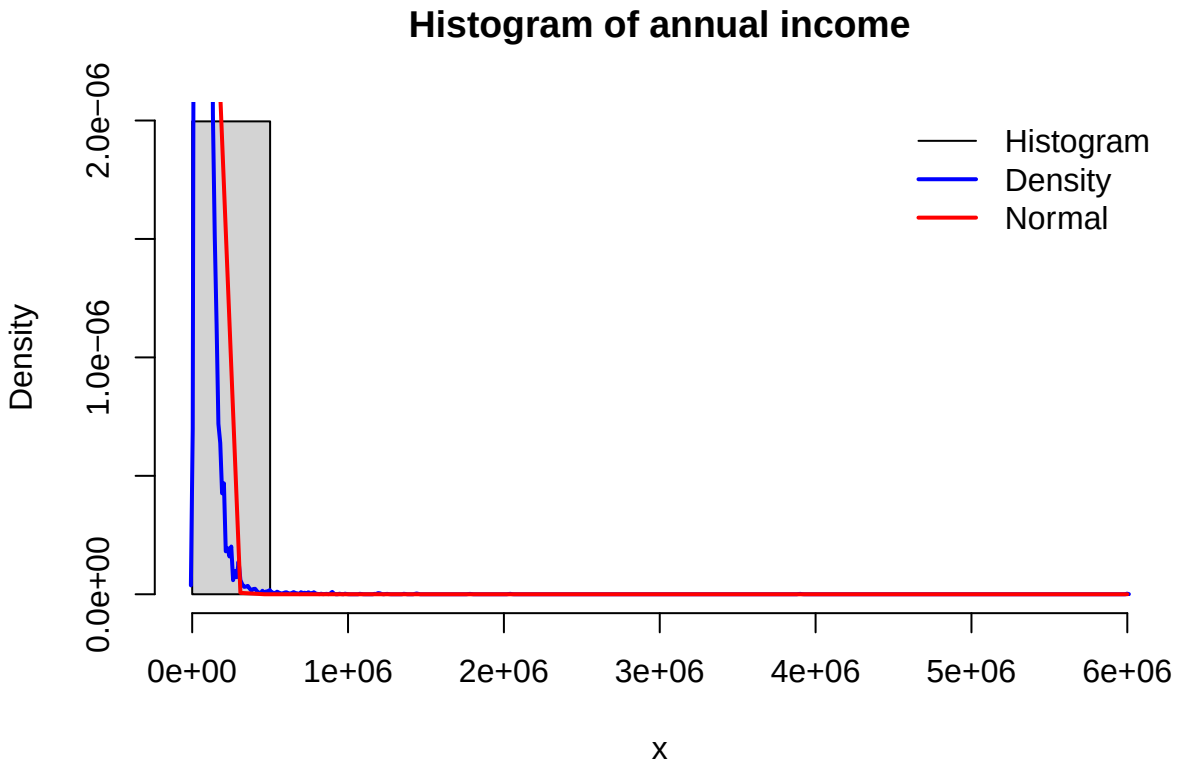
```
# Histograms for continuous varibles: loan_amnt, int_rate, annual_income, dti
histDenNorm <- function(x, main = "") {
    hist(x, prob = TRUE, main = main) # Histogram
    lines(density(x), col = "blue", lwd = 2) # Density
    x2 <- seq(min(x), max(x), length = 40)
    f <- dnorm(x2, mean(x), sd(x))
    lines(x2, f, col = "red", lwd = 2) # Normal
    legend("topright", c("Histogram", "Density", "Normal"),
        box.lty = 0,
        lty = 1, col = c("black", "blue", "red"), lwd = c(1, 2, 2)
    )
}

histDenNorm(df4$loan_amnt, main = "Histogram of loan_amount")
```



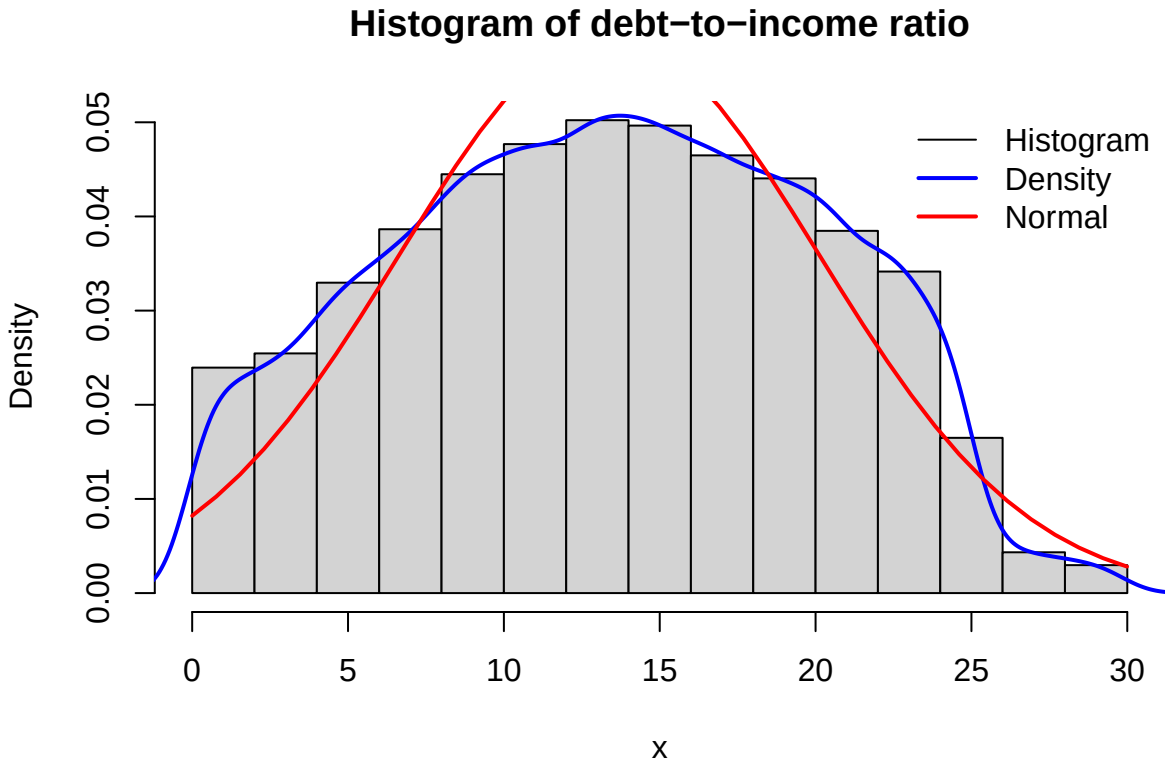

```
histDenNorm(df4$int_rate, main = "Histogram of interest rate")
```



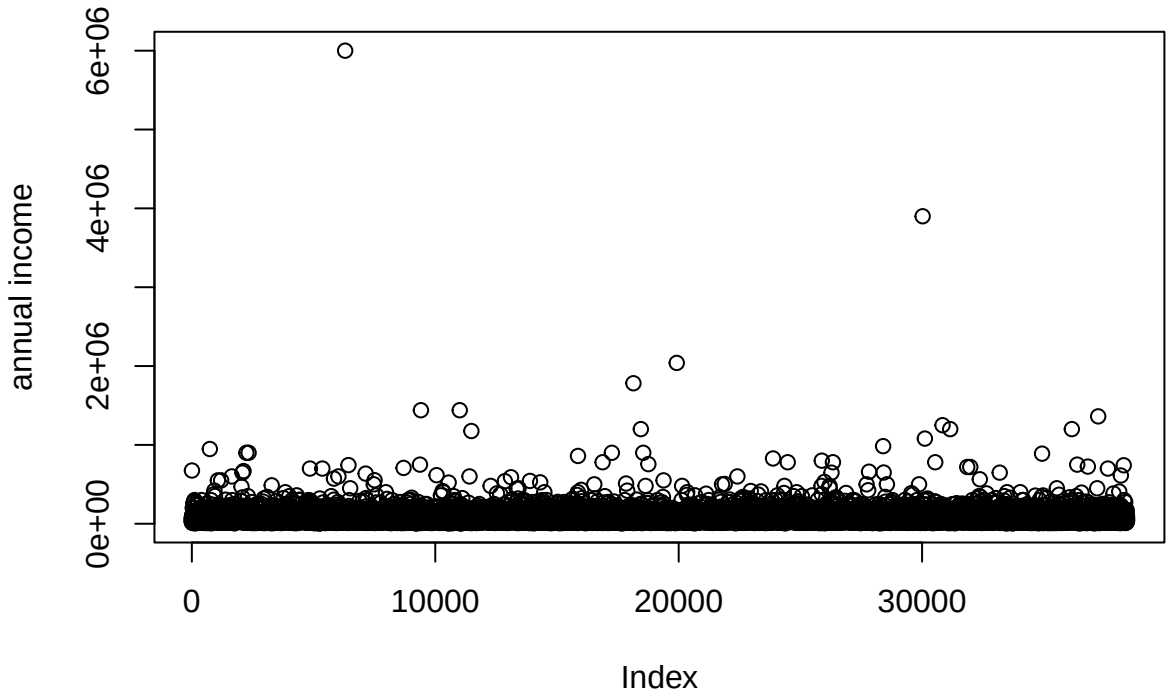

```
histDenNorm(df4$annual_inc, main = "Histogram of annual income")
```




```
histDenNorm(df4$dti, main = "Histogram of debt-to-income ratio")
```

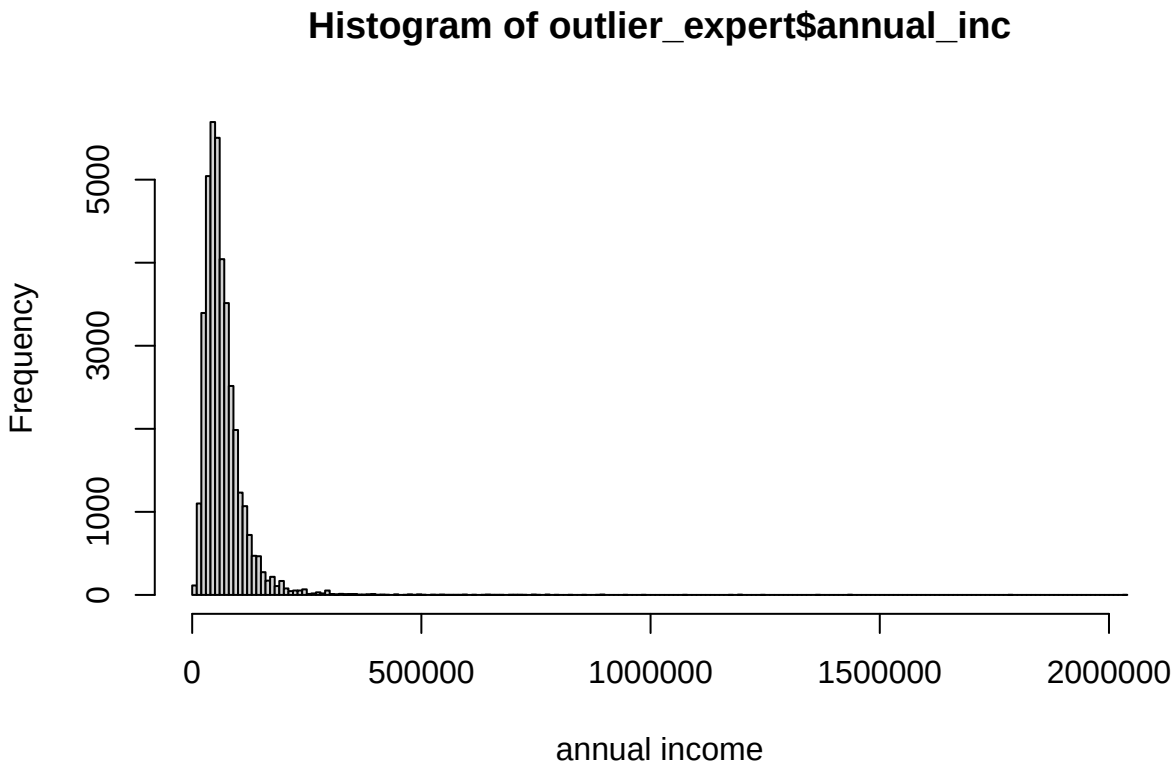



```
plot(df4$annual_inc, ylab = "annual income")
```

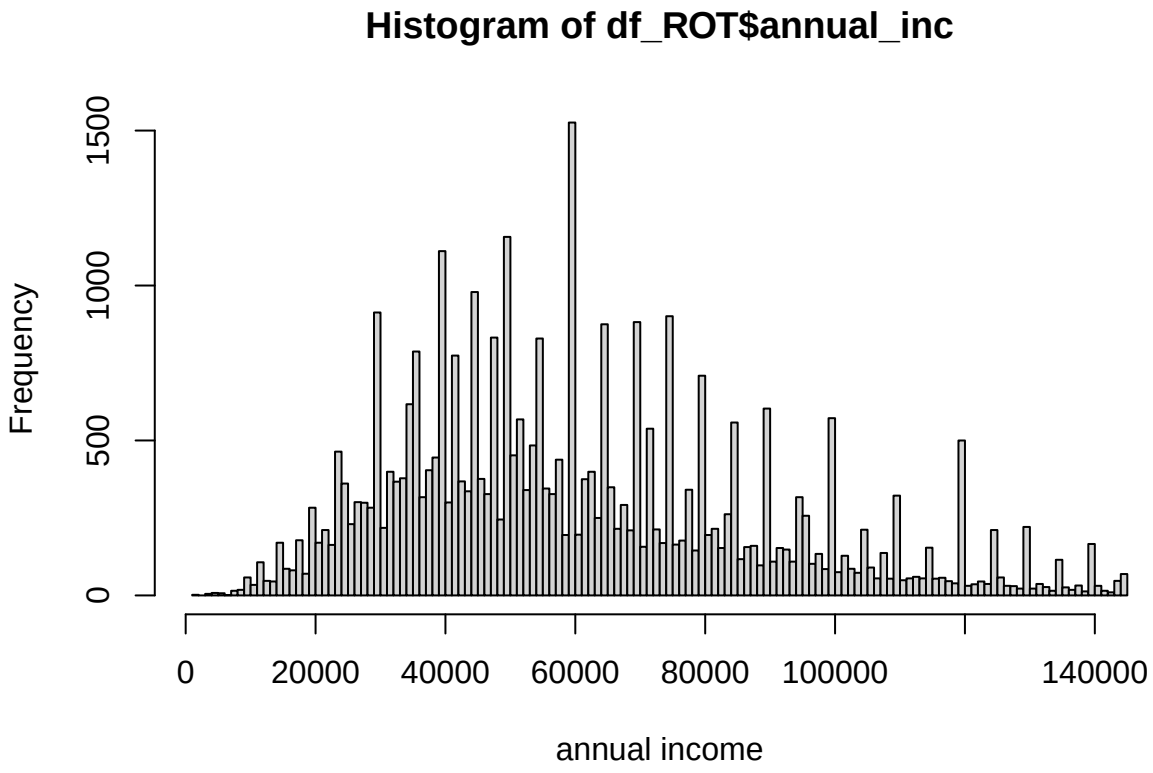


Hence, we need to remove values above 3 million dollars. That's an outlier. we can remove outlier using rule of thumb and keep the range between first and third quartile.

```
index_outlier <- which(df4$annual_inc > 3000000)
outlier_expert <- df4[-index_outlier, ]

outlier_cutoff <- quantile(df4$annual_inc, 0.75) + 1.5 * IQR(df4$annual_inc)
index_outlier_ROT <- which(df4$annual_inc > outlier_cutoff)
df_ROT <- df4[-index_outlier_ROT, ]


hist(outlier_expert$annual_inc, sqrt(nrow(outlier_expert)), xlab = "annual income")
```

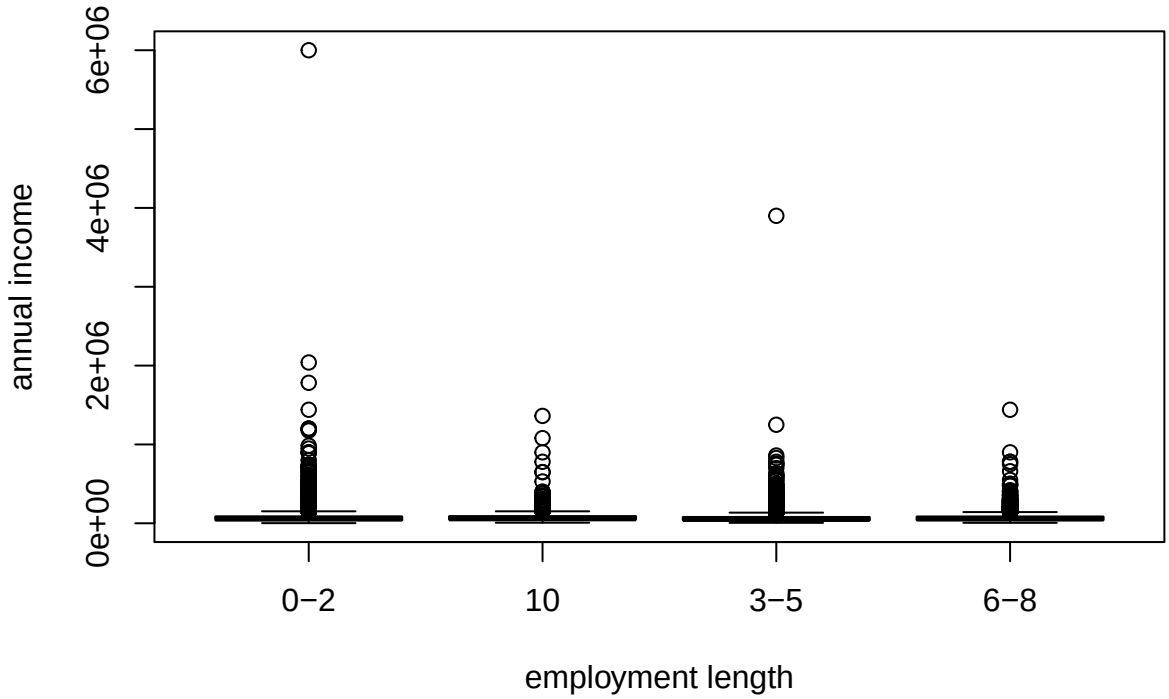



```
hist(df_ROT$annual_inc, sqrt(nrow(df_ROT)), xlab = "annual income")
```

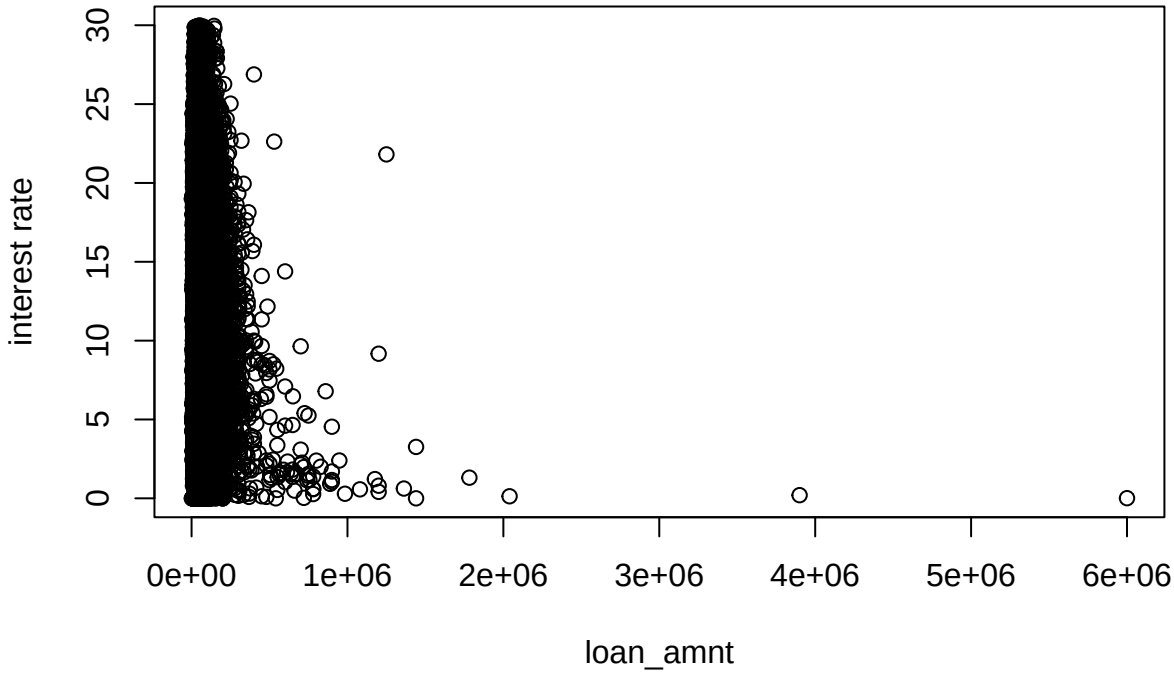


Bivariate variables analysis

```
plot(df4$emp_length, df4$annual_inc, xlab = "employment length", ylab = "annual income")
```




```
plot(df4$annual_inc, df4$dti, xlab = "loan_amnt", ylab = "interest rate")
```




```
df5 <- df4[!(df4$annual_inc > 140000), ]#as rule of thumb suggests
df5 <- df4[, !(colnames(df4) %in% c("purpose"))]

#remove loan amounts higher than 3000000
```

```
df5 <- df5[!(df5$loan_amnt > 3000000), ]#as scatter plot  suggests
```

Analysis with linear model

```
pda_full <- lm(repay_fail ~ loan_amnt + term + int_rate +
  emp_length + home_ownership + annual_inc + loan_status + dti +
  delinq_2yrs + inq_last_6mths + pub_rec + revol_util, data = df5)
# summary(pda_full)

pda1 <- lm(repay_fail ~ loan_amnt + term + int_rate + emp_length +
  home_ownership + annual_inc + loan_status + dti + inq_last_6mths + pub_rec, data = df5)
# summary(pda1)

pda2 <- lm(repay_fail ~ loan_amnt + term + int_rate + emp_length +
  home_ownership + annual_inc + dti + inq_last_6mths + pub_rec, data = df5)
#summary(pda2)

extractAIC(pda_full)
```

```
## [1]       17 -2599155
```

```
extractAIC(pda1)
```

```
## [1]       15 -2599159
```

```
extractAIC(pda2)
```

```
## [1]     14.00 -80876.99
```

```
ols_step_both_aic(pda_full)
```

```
##
##
##                                  Stepwise Summary
## -------------------------------------------------------------------------------------
## Variable        Method          AIC           RSS       Sum Sq      R-Sq       Adj. R-Sq
## -------------------------------------------------------------------------------------
## loan_status    addition    -2173587.709    0.000    4929.952    1.00000       1.00000
## -------------------------------------------------------------------------------------
```

```
ols_step_both_aic(pda1)
```

```
##
##
##                                  Stepwise Summary
## -------------------------------------------------------------------------------------
## Variable        Method          AIC           RSS       Sum Sq      R-Sq       Adj. R-Sq
## -------------------------------------------------------------------------------------
## loan_status    addition    -2173587.709    0.000    4929.952    1.00000       1.00000
## -------------------------------------------------------------------------------------
```

```
ols_step_both_aic(pda2)
```

```
##
##
```

```
##                                 Stepwise Summary
## ----------------------------------------------------------------------------------------
## Variable            Method        AIC         RSS       Sum Sq      R-Sq      Adj. R-Sq
## ----------------------------------------------------------------------------------------
## int_rate          addition    28571.122    4731.224    198.728    0.04031       0.04029
## term              addition    28457.184    4716.967    212.984    0.04320       0.04315
## inq_last_6mths    addition    28331.746    4701.346    228.605    0.04637       0.04630
## annual_inc        addition    28232.893    4689.021    240.931    0.04887       0.04877
## pub_rec           addition    28181.511    4682.510    247.442    0.05019       0.05007
## emp_length        addition    28167.034    4680.015    249.937    0.05070       0.05050
## home_ownership    addition    28159.542    4678.372    251.580    0.05103       0.05076
## loan_amnt         addition    28154.951    4677.569    252.383    0.05119       0.05090
## dti               addition    28150.570    4676.792    253.159    0.05135       0.05103
## ----------------------------------------------------------------------------------------
```

```
# Final Dataset for Model building #Final Dataset for Model building: Take significant variables into t

final_datas <- df5[, !(colnames(df5) %in% c("loan_status", "loan_amnt", "delinq_2yrs", "revol_util"))]
final_datas
```

```
## # A tibble: 38,418 x 9
##    term  int_rate emp_length home_ownership annual_inc   dti inq_last_6mths
##    <fct>    <dbl> <fct>      <fct>               <dbl> <dbl> <fct>
##  1 "36 "    14.0  3-5        2                   20004 19.9  0
##  2 "36 "    16.0  3-5        2                   59000 19.6  0
##  3 "36 "     9.91 0-2        1                   53796 10.8  0
##  4 "36 "     5.42 0-2        2                   30000  3.6  0
##  5 "36 "    10.2  0-2        1                  675048  1.55 0
##  6 "36 "     6.03 0-2        1                   77736  6.07 0
##  7 "36 "     7.49 3-5        2                   35000 13.1  0
##  8 "60 "    14.3  3-5        2                   86000 26.5  0
##  9 "60 "    23.2  0-2        1                   72500 20.0  0
## 10 "36 "    17.3  3-5        1                   28000 13.8  0
## # ... with 38,408 more rows, and 2 more variables: pub_rec <fct>,
## #   repay_fail <dbl>
```

Using both side AIC fuction we received summary of important variables and using them we will create our binomial logistic regression model. According to the data explotory analysises, loan_status, delinq_2yrs and revol_util are considered as insignificant, so we have selected loan_amnt, term, int_rate, emp_length, home_ownership, annual_inc, dti, inq_last_6mths, pub_rec for further default prediction task. With these variables, final dataset has been prepared.

Modelling

Bernoulli distribution GLM Logistic regression with Logit link function. Training the Logistic Regression and check statistics.

Data split

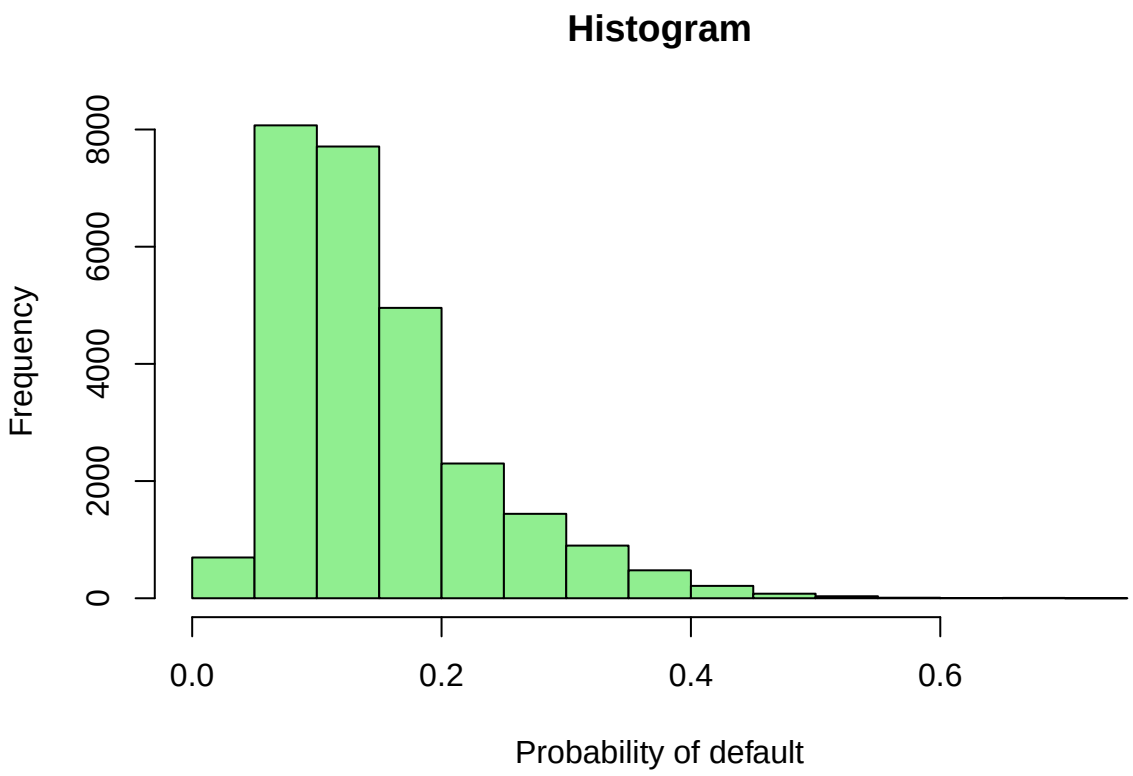
```
data_split <- createDataPartition(final_datas$repay_fail, p = .70, list = FALSE)
train <- final_datas[data_split, ]
test <- final_datas[-data_split, ]
```

Training the Logistic Regression and check statistics

```
#using logit link function

logit.fit <- glm(repay_fail ~ . , data = train, family = binomial(link = "logit"))
summary(logit.fit)
```

```
##
## Call:
## glm(formula = repay_fail ~ ., family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4723  -0.5954  -0.4806  -0.3675   3.2026
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -3.237e+00  1.006e-01 -32.189  < 2e-16 ***
## term60           3.768e-01  4.199e-02   8.973  < 2e-16 ***
## int_rate         1.289e-01  5.526e-03  23.319  < 2e-16 ***
## emp_length10    -4.536e-02  7.268e-02  -0.624 0.532565
## emp_length3-5   -1.449e-01  4.232e-02  -3.424 0.000618 ***
## emp_length6-8   -5.135e-02  4.855e-02  -1.058 0.290256
## home_ownership1 -3.708e-02  6.958e-02  -0.533 0.594126
## home_ownership2 -6.749e-03  6.798e-02  -0.099 0.920920
## home_ownership3  5.891e-01  2.760e-01   2.134 0.032830 *
## annual_inc      -4.725e-06  4.986e-07  -9.476  < 2e-16 ***
## dti              3.717e-03  2.677e-03   1.388 0.165006
## inq_last_6mths1  8.305e-01  1.003e-01   8.283  < 2e-16 ***
## pub_rec1         3.754e-01  6.659e-02   5.638 1.72e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22537  on 26892  degrees of freedom
## Residual deviance: 21201  on 26880  degrees of freedom
## AIC: 21227
##
## Number of Fisher Scoring iterations: 5
```

```
#This will make predictions on the training data that you use to fit the model and give me a vector of

logit.probs <- predict(logit.fit,
                       newdata = test,
                       type = "response")

hist(logit.fit$fitted.values,main = " Histogram ",xlab = "Probability of default", col = 'light green')
```




```
logit.pred <- ifelse(logit.probs > 0.20, "1", "0")

repay_fail.logit = test$repay_fail
logit_table <- table(repay_fail.logit, logit.pred)
logit_table
```

```
##                 logit.pred
## repay_fail.logit    0    1
##                0 7977 1717
##                1 1110  721
```

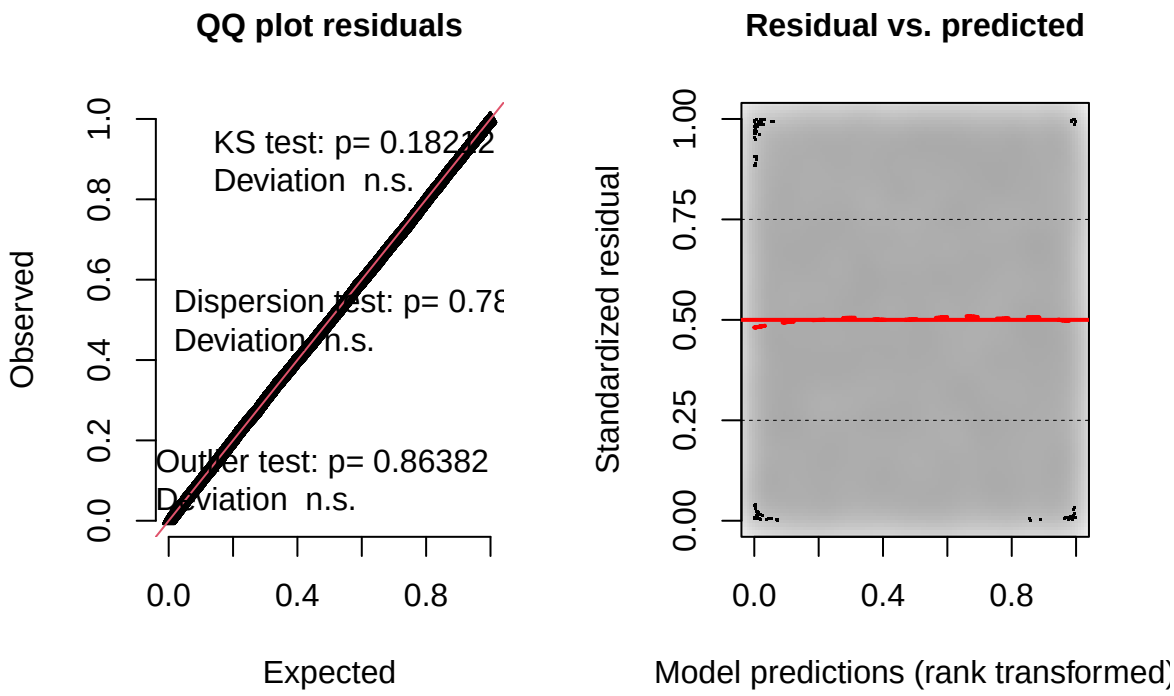
```
logit_mean <- mean(logit.pred == repay_fail.logit)
logit_mean
```

```
## [1] 0.7547072
```

```
Gini(logit.pred, test$repay_fail)
```

```
## [1] 0.2019524
```

```
#Residuals
res.logit = simulateResiduals(logit.fit)
plot(res.logit)
```



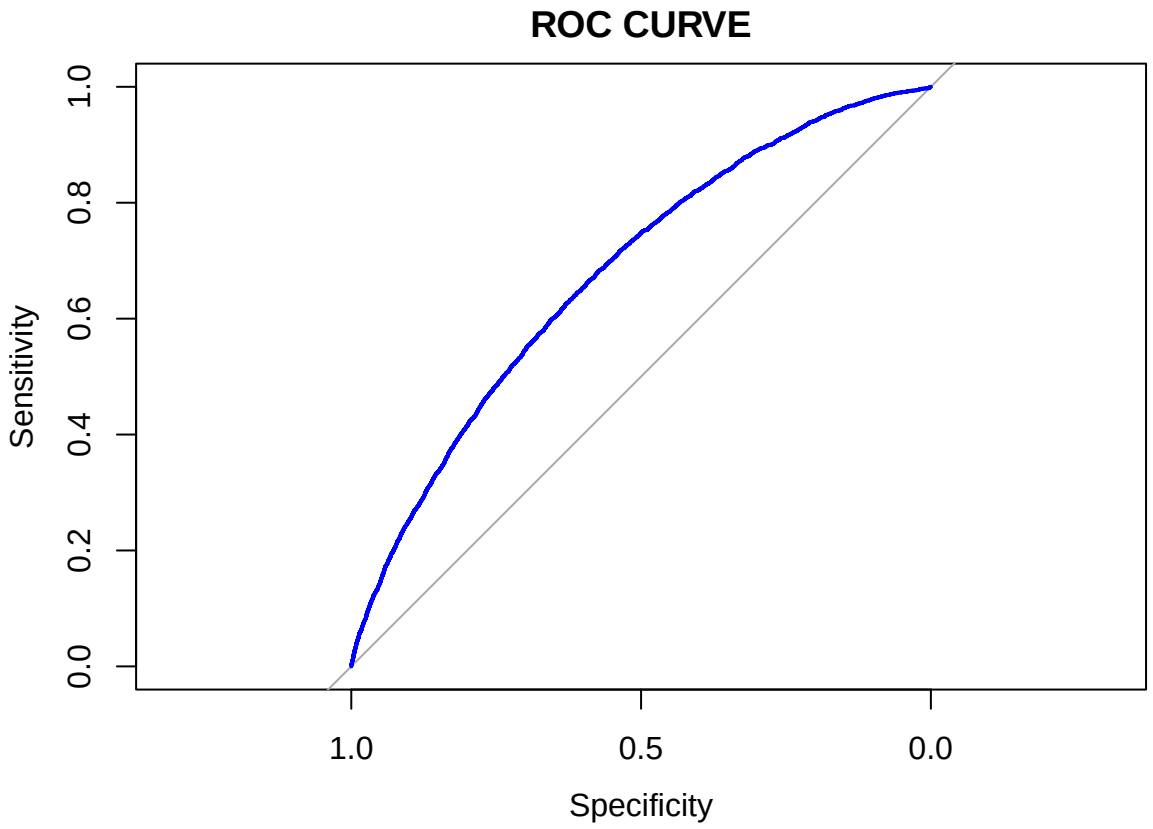

```
#Model Performance:
logit.fit$aic
```

```
## [1] 21226.93
```

```
#Roc curve
roc(repay_fail ~ logit.fit$fitted.values, data = train, plot = TRUE, main = "ROC CURVE", col= "blue")
```

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

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




```
##
## Call:
## roc.formula(formula = repay_fail ~ logit.fit$fitted.values, data = train,     plot = TRUE, main = "R
##
## Data: logit.fit$fitted.values in 22916 controls (repay_fail 0) < 3977 cases (repay_fail 1).
## Area under the curve: 0.6774
```

```
#AUC under the curve
auc(repay_fail~logit.fit$fitted.values, data = train)
```

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

```
## Area under the curve: 0.6774
```

```
#using probit link function

probit.fit <- glm(repay_fail ~ . , data = train, family = binomial(link = "probit"))
```

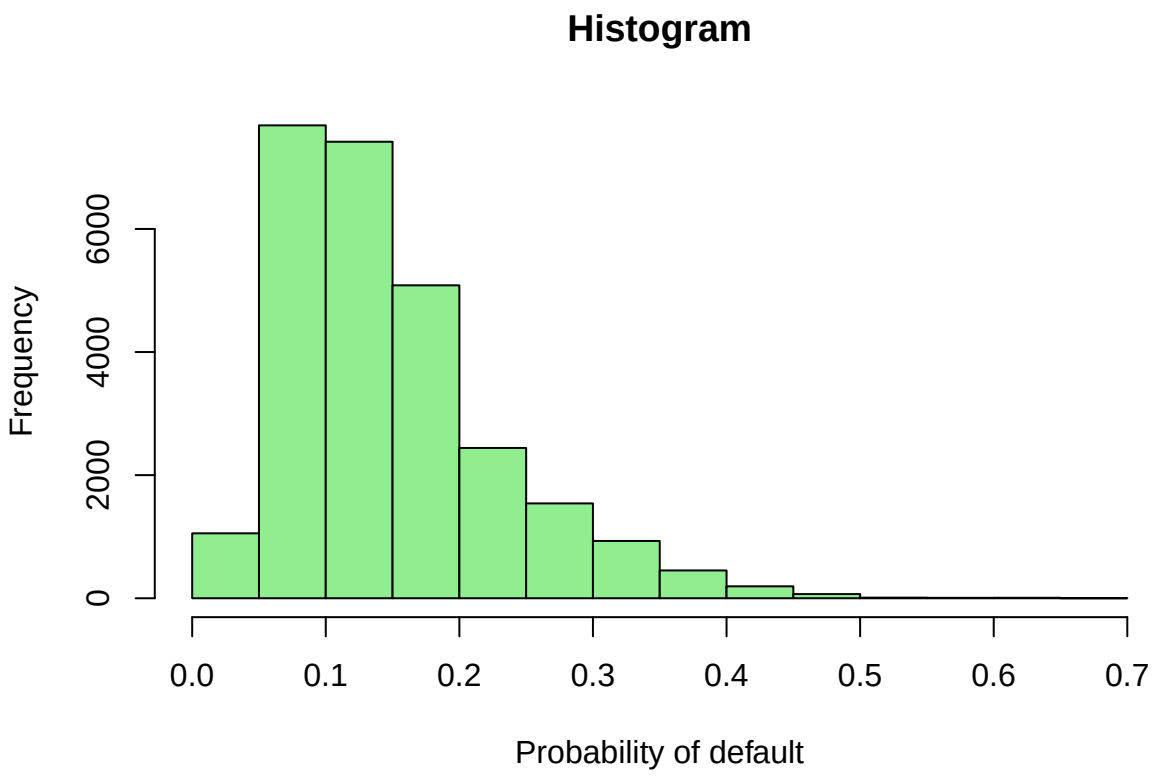
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(probit.fit)
```

```
##
## Call:
## glm(formula = repay_fail ~ ., family = binomial(link = "probit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -1.4232  -0.6002  -0.4822  -0.3584   3.3030
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.883e+00  5.421e-02 -34.732  < 2e-16 ***
## term60           2.129e-01  2.346e-02   9.075  < 2e-16 ***
## int_rate         7.136e-02  3.015e-03  23.671  < 2e-16 ***
## emp_length10    -2.369e-02  3.976e-02  -0.596 0.551272
## emp_length3-5   -7.902e-02  2.310e-02  -3.422 0.000622 ***
## emp_length6-8   -2.535e-02  2.665e-02  -0.951 0.341368
## home_ownership1 -2.460e-02  3.813e-02  -0.645 0.518851
## home_ownership2 -2.783e-03  3.737e-02  -0.074 0.940633
## home_ownership3  3.271e-01  1.605e-01   2.039 0.041485 *
## annual_inc      -2.246e-06  2.571e-07  -8.737  < 2e-16 ***
## dti              2.114e-03  1.464e-03   1.444 0.148752
## inq_last_6mths1  4.899e-01  6.016e-02   8.143 3.86e-16 ***
## pub_rec1         2.116e-01  3.838e-02   5.513 3.52e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22537  on 26892  degrees of freedom
## Residual deviance: 21193  on 26880  degrees of freedom
## AIC: 21219
##
## Number of Fisher Scoring iterations: 6
```

```r
#This will make predictions on the training data that you use to fit the model and give me a vector of

pro.probs <- predict(probit.fit,
                     newdata = test,
                     type = "response")

hist(probit.fit$fitted.values,main = " Histogram ",xlab = "Probability of default", col = 'light green'
```




```
pro.pred <- ifelse(pro.probs > 0.20, "1", "0")

repay_fail.prob = test$repay_fail
probit_table <- table(repay_fail.prob, pro.pred)
probit_table
```

```
##                pro.pred
## repay_fail.prob    0    1
##               0 7915 1779
##               1 1086  745
```

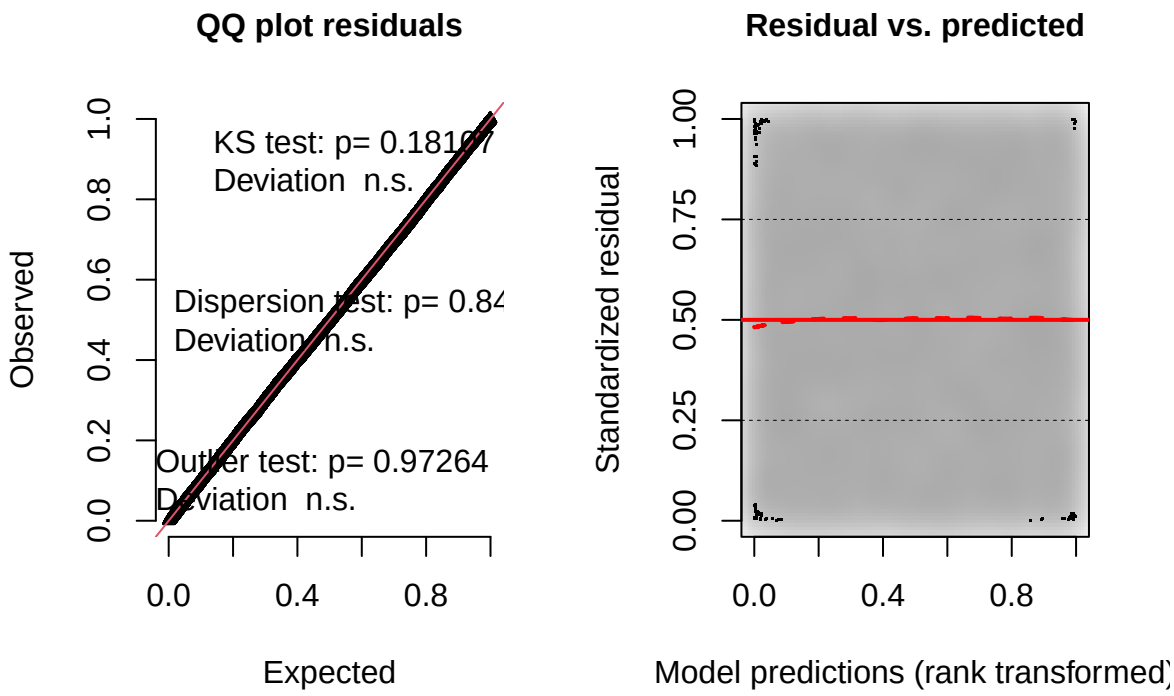
```
probit_mean <- mean(pro.pred == repay_fail.prob)
probit_mean
```

```
## [1] 0.75141
```

```
Gini(pro.pred, test$repay_fail)
```

```
## [1] 0.2097659
```

```
#Residuals
res.probit = simulateResiduals(probit.fit)
plot(res.probit)
```



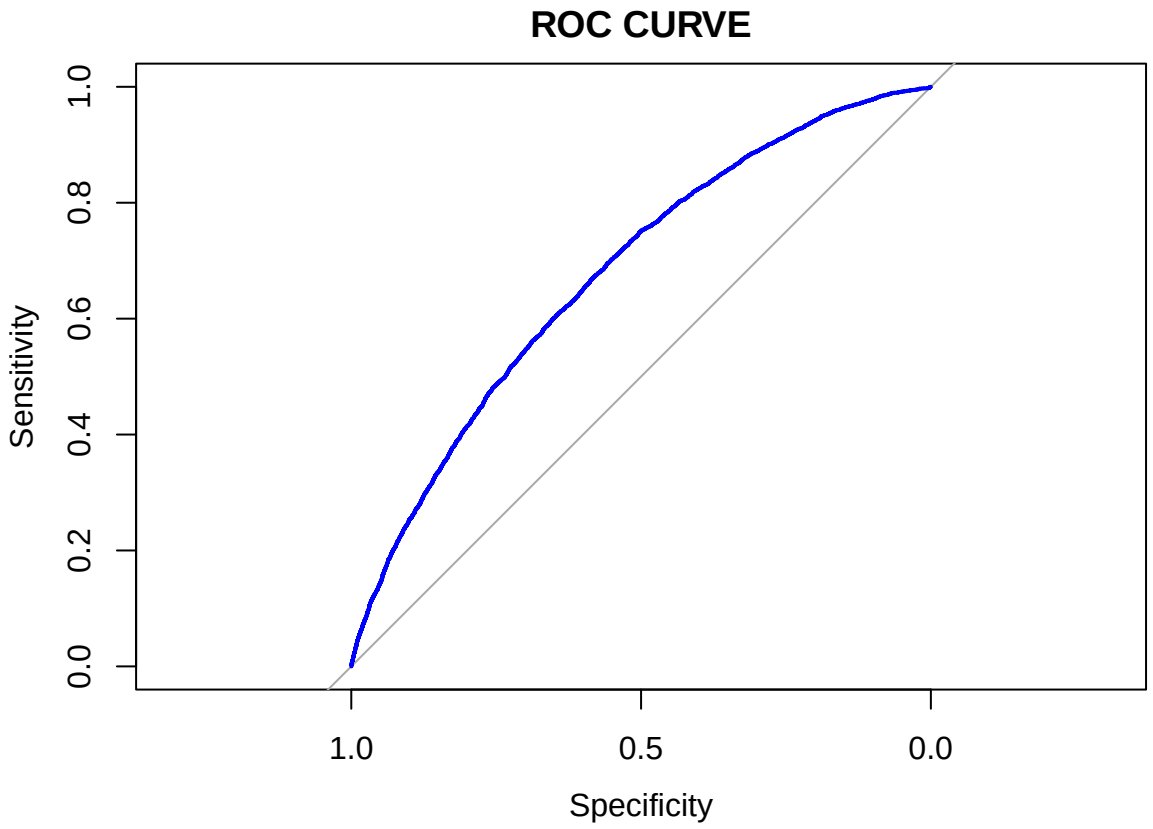

```
#Model Performance:
probit.fit$aic
```

```
## [1] 21219.5
```

```
#Roc curve
roc(repay_fail ~ probit.fit$fitted.values, data = train, plot = TRUE, main = "ROC CURVE", col= "blue")
```

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

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




```
##
## Call:
## roc.formula(formula = repay_fail ~ probit.fit$fitted.values,     data = train, plot = TRUE, main = "
##
## Data: probit.fit$fitted.values in 22916 controls (repay_fail 0) < 3977 cases (repay_fail 1).
## Area under the curve: 0.6772
```

```
#AUC under the curve
auc(repay_fail~probit.fit$fitted.values, data = train)
```

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

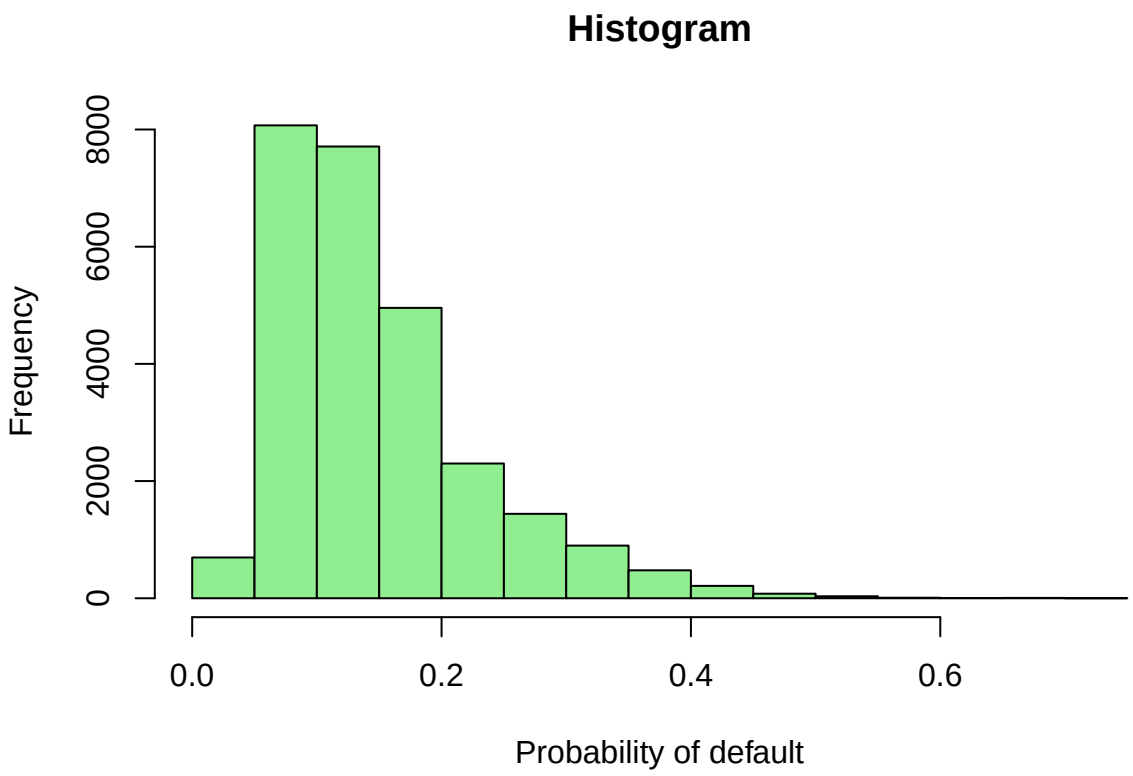
```
## Area under the curve: 0.6772
```

```
#using cloglog link function

clog.fit <- glm(repay_fail ~ . , data = train, family = binomial(link = "logit"))
summary(clog.fit)
```

```
##
## Call:
## glm(formula = repay_fail ~ ., family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4723  -0.5954  -0.4806  -0.3675   3.2026
##
## Coefficients:
```

```
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -3.237e+00  1.006e-01 -32.189  < 2e-16 ***
## term60           3.768e-01  4.199e-02   8.973  < 2e-16 ***
## int_rate         1.289e-01  5.526e-03  23.319  < 2e-16 ***
## emp_length10    -4.536e-02  7.268e-02  -0.624 0.532565
## emp_length3-5   -1.449e-01  4.232e-02  -3.424 0.000618 ***
## emp_length6-8   -5.135e-02  4.855e-02  -1.058 0.290256
## home_ownership1 -3.708e-02  6.958e-02  -0.533 0.594126
## home_ownership2 -6.749e-03  6.798e-02  -0.099 0.920920
## home_ownership3  5.891e-01  2.760e-01   2.134 0.032830 *
## annual_inc      -4.725e-06  4.986e-07  -9.476  < 2e-16 ***
## dti              3.717e-03  2.677e-03   1.388 0.165006
## inq_last_6mths1  8.305e-01  1.003e-01   8.283  < 2e-16 ***
## pub_rec1         3.754e-01  6.659e-02   5.638 1.72e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22537  on 26892  degrees of freedom
## Residual deviance: 21201  on 26880  degrees of freedom
## AIC: 21227
##
## Number of Fisher Scoring iterations: 5
```

```r
#This will make predictions on the training data that you use to fit the model and give me a vector of

clog.probs <- predict(clog.fit,
                      newdata = test,
                      type = "response")

hist(clog.fit$fitted.values,main = " Histogram ",xlab = "Probability of default", col = 'light green')
```




```
clog.pred <- ifelse(clog.probs > 0.20, "1", "0")

repay_fail.clog = test$repay_fail
clog_table <- table(repay_fail.clog, clog.pred)
clog_table
```

```
##                clog.pred
## repay_fail.clog    0    1
##               0 7977 1717
##               1 1110  721
```

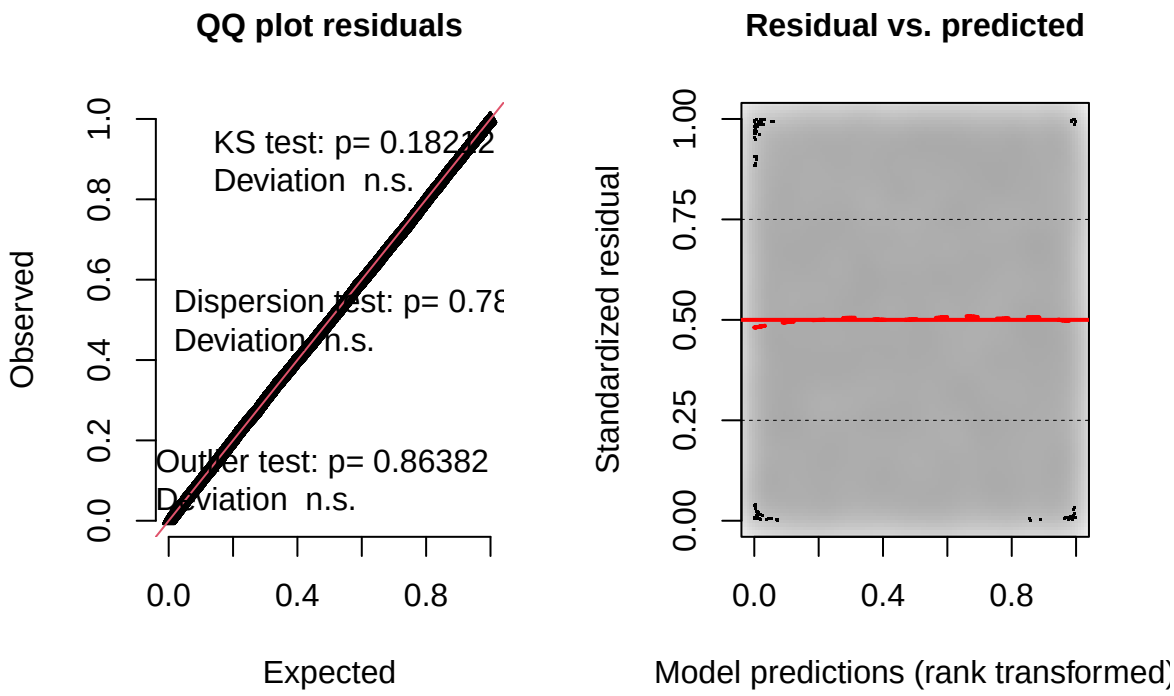
```
clog_mean <- mean(clog.pred == repay_fail.clog)
clog_mean
```

```
## [1] 0.7547072
```

```
Gini(clog.pred, test$repay_fail)
```

```
## [1] 0.2019524
```

```
#Residuals
res.clog = simulateResiduals(clog.fit)
plot(res.clog)
```



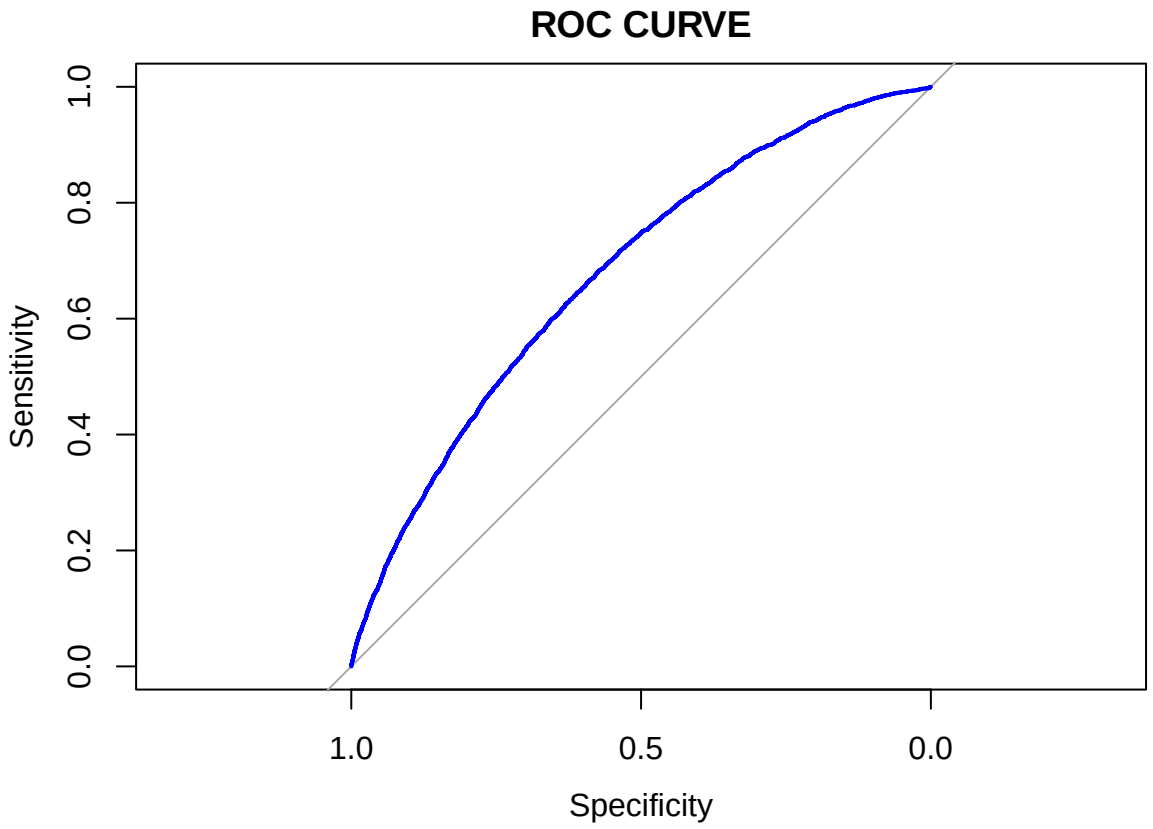

```
#Model Performance:
clog.fit$aic
```

```
## [1] 21226.93
```

```
#Roc curve
roc(repay_fail ~ clog.fit$fitted.values, data = train, plot = TRUE, main = "ROC CURVE", col= "blue")
```

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

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




```
##
## Call:
## roc.formula(formula = repay_fail ~ clog.fit$fitted.values, data = train,     plot = TRUE, main = "RO
##
## Data: clog.fit$fitted.values in 22916 controls (repay_fail 0) < 3977 cases (repay_fail 1).
## Area under the curve: 0.6774
```

```
#AUC under the curve
auc(repay_fail~clog.fit$fitted.values, data = train)
```

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

```
## Area under the curve: 0.6774
```

#Gini values and accuracy shows not much of difference in variables however it does shows that logit function is best as it is easy tointerpret. And other significant link function does'nt show significant observation. Also, our default rate is kept 20% higher than 15 % which increase predictive capability. All variables are of significance in the model.Also showcase that employment length less than 10 years has significance in default rates also less than 5 years have higher significance. Rental owners have highest significance to default opposite to that none other category shows any significance. dti shows mild significance and hence considered. The Roc curve defines specificity to sensitivity. And compared to older model. Our curve converges more onto top-left corner which shows better performance. Our model gini value is 0.1938986. This is higher compared to old model i.e. 0.114; this shows dispersion is higher than older model.

Build model with Cross-Validation

```r
#Cross-validation to induce over-fitting
#factorise dependent variable

#Data Split

data_split <- createDataPartition(final_datas$repay_fail, p = .70, list = FALSE)
train1 <- final_datas[data_split, ]
test1 <- final_datas[-data_split, ]

#add labels to response variable (1= default and 0 = non-default)
train1$repay_fail[train1$repay_fail==1] <- "default"
train1$repay_fail[train1$repay_fail==0] <- "nondefault"

test1$repay_fail[test1$repay_fail==1] <- "default"
test1$repay_fail[test1$repay_fail==0] <- "nondefault"

#Covert Outcome Variable to type factor
train1$repay_fail <- as.factor(train1$repay_fail)
test1$repay_fail <- as.factor(test1$repay_fail)

# Define training control

ctrlspecs <- trainControl(
    method = "CV", number = 10,
    savePredictions = "all",
    classProbs = TRUE
)

#set random seed
set.seed(125)

# Train the model
model1 <- train(repay_fail~ .,
                data = train1,
                method = "glm", family = binomial,
                trControl = ctrlspecs)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
print(model1)
```

```
## Generalized Linear Model
##
## 26893 samples
##     8 predictor
##     2 classes: 'default', 'nondefault'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 24203, 24204, 24203, 24203, 24205, 24204, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8483992  0.009098461
```

```
summary(model1)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.9463   0.3658   0.4825   0.6037   1.5226
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      3.145e+00  9.914e-02  31.724  < 2e-16 ***
## `term60 `       -3.651e-01  4.157e-02  -8.783  < 2e-16 ***
## int_rate        -1.309e-01  5.479e-03 -23.888  < 2e-16 ***
## emp_length10     6.480e-02  7.255e-02   0.893   0.3718
## `emp_length3-5`  1.938e-01  4.196e-02   4.618 3.88e-06 ***
## `emp_length6-8`  9.589e-02  4.843e-02   1.980   0.0477 *
## home_ownership1  1.199e-01  6.835e-02   1.754   0.0795 .
## home_ownership2  5.188e-02  6.657e-02   0.779   0.4357
## home_ownership3 -3.360e-01  2.793e-01  -1.203   0.2290
## annual_inc       5.207e-06  5.065e-07  10.281  < 2e-16 ***
## dti             -5.322e-03  2.664e-03  -1.997   0.0458 *
## inq_last_6mths1 -8.870e-01  9.892e-02  -8.967  < 2e-16 ***
## pub_rec1        -4.027e-01  6.639e-02  -6.066 1.31e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22850  on 26892  degrees of freedom
## Residual deviance: 21432  on 26880  degrees of freedom
## AIC: 21458
##
## Number of Fisher Scoring iterations: 5
```

Statistically significant

```
varImp(model1)
```

Variable importance (predictor variables)

```
## glm variable importance
##
##                 Overall
## int_rate        100.000
## annual_inc       41.116
## inq_last_6mths1  35.432
## `term60 `        34.633
## pub_rec1         22.876
## `emp_length3-5`  16.610
## dti               5.271
## `emp_length6-8`   5.195
## home_ownership1   4.215
## home_ownership3   1.833
## emp_length10      0.492
## home_ownership2   0.000
```

According to result, the most importance variables are int_rate, annual_inc, term60, inq_last_6mths1, pub_rec1, emp_length3-5, dti, emp_length6-8 and emp_length10

Apply model to test1

```
# predict outcome using test dataset
predictn <- predict(model1, newdata = test1)
```

Confusion Matrix

```
confusionMatrix(data = predictn, test1$repay_fail)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   default nondefault
##   default         21         23
##   nondefault    1720       9761
##
##                Accuracy : 0.8488
##                  95% CI : (0.8421, 0.8553)
##     No Information Rate : 0.8489
##     P-Value [Acc > NIR] : 0.5271
##
##                   Kappa : 0.0162
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.012062
##             Specificity : 0.997649
##          Pos Pred Value : 0.477273
##          Neg Pred Value : 0.850187
```

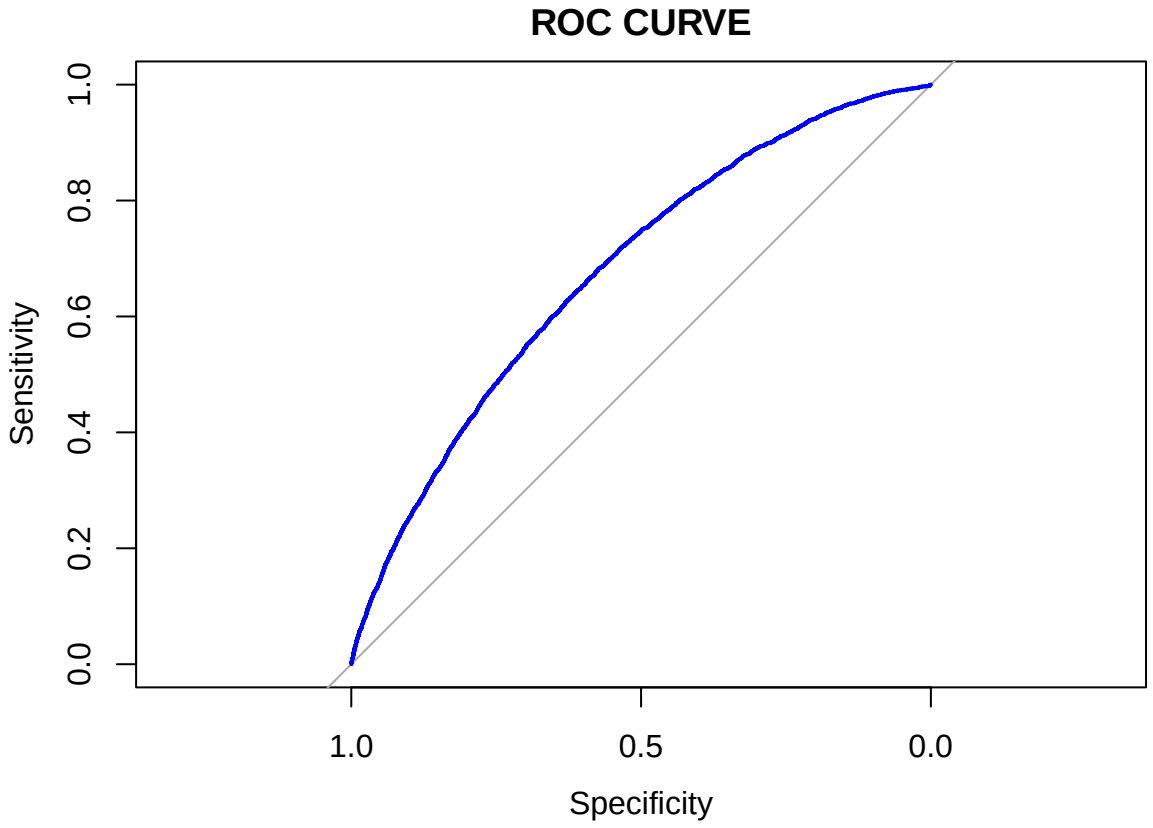
```
##              Prevalence : 0.151063
##          Detection Rate : 0.001822
##    Detection Prevalence : 0.003818
##       Balanced Accuracy : 0.504856
##
##        'Positive' Class : default
##
```

Comparing model with ROC Curve

```
#Roc curve
roc(repay_fail ~ clog.fit$fitted.values, data = train, plot = TRUE, main = "ROC CURVE", col= "blue")
```

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

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




```
##
## Call:
## roc.formula(formula = repay_fail ~ clog.fit$fitted.values, data = train,     plot = TRUE, main = "RO
##
## Data: clog.fit$fitted.values in 22916 controls (repay_fail 0) < 3977 cases (repay_fail 1).
## Area under the curve: 0.6774
```

```
#AUC under the curve
auc(repay_fail~clog.fit$fitted.values, data = train)
```

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

```
## Area under the curve: 0.6774
```

Our model gini value is 0.1938986. This is higher compared to old model i.e. 0.114; this shows dispersion is higher than older model. In our cross validation model, gini value is 0.1938986. This is higher compared to old model i.e. 0.114; this shows dispersion is higher than older model.