Exploratory Data Analysis

Classification & Prediction of Loan Defaulters

- Multivariate-Analysis-Project Repository Link:
  - https://github.com/richardbritto97/Multivariate-Analysis-Project
- Team 8:
  - Adarsh Lalchandani https://github.com/AdarshRL2109
  - Karthik Grandhi https://github.com/karthii24
  - Richard Britto https://github.com/richardbritto97

Loading Necessary Packages & Libraries

```r
library(data.table)
library(magrittr)
library(stringr)
library(ggplot2)
library(knitr)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```r
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------ tidyverse 1.3.0 --

## v tibble  3.0.3     v purrr   0.3.4
## v tidyr   1.1.2     v dplyr   1.0.2
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts --------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::set_names() masks magrittr::set_names()
## x purrr::transpose() masks data.table::transpose()
```

Data Loading

Importing the csv format of our Lending Dataset

The data dictionary:

Variable	Description
member_id	Unique identifier
loan_status	Current status of the loan
int_rate	Interest Rate on the loan
Bin_int	int_rate bins for categorical use
dti	A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income.
Bin_dti	dti bins for categorical use
Default_flag	A Boolean value where 0 means no default & 1 means default
No_of_Enquiry	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
enq_buckets	bucket or groups of enquiry for categorical use
annual_inc	The self-reported annual income provided by the borrower during registration.
Income_bins	bins of income for categorical use and to map outliers easily
Purpose	A category provided by the borrower for the loan request.
home_ownership	status of the ownership of the borrower's property, takes categorical value
purpose	states the purpose for which the loan was taken
open_acc	The number of open credit lines in the borrower's credit file.
emp_length	The job title supplied by the Borrower when applying for the loan.
verification_status	the status of verification stating whether source verified, verified, or not verified
delinq_2yrs	The past-due amount owed for the accounts on which the borrower is now delinquent.
loan_amnt	The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.
Bins_loan_amt	Bins for loan_amnt for categorical use

```r
Lending_Data <- read_csv('Lending_Data.csv')
```

```
## Parsed with column specification:
## cols(
##   member_id = col_character(),
##   loan_status = col_character(),
##   int_rate = col_character(),
##   Bin_int = col_double(),
##   dti = col_double(),
##   Bin_dti = col_double(),
##   Default_flag = col_double(),
##   No_of_Enquiry = col_double(),
##   enq_buckets = col_character(),
```

```
##   annual_inc = col_double(),
##   Income_bins = col_double(),
##   home_ownership = col_character(),
##   purpose = col_character(),
##   open_acc = col_double(),
##   emp_length = col_character(),
##   verification_status = col_character(),
##   delinq_2yrs = col_double(),
##   loan_amnt = col_double(),
##   Bins_loan_amt = col_double()
## )
```

```r
Lend = copy(Lending_Data)
Lend = setDT(Lend)
view(Lend)
str(Lend)
```

```
## Classes 'data.table' and 'data.frame':   35808 obs. of  19 variables:
##  $ member_id          : chr  "LC1" "LC10" "LC100" "LC1000" ...
##  $ loan_status        : chr  "Charged Off" "Fully Paid" "Fully Paid" "Fully Paid" ...
##  $ int_rate           : chr  "11.71%" "15.96%" "10.65%" "12.69%" ...
##  $ Bin_int            : num  10 16 8 11 22 1 23 10 5 16 ...
##  $ dti                : num  1.06 2.61 11.34 14 13.01 ...
##  $ Bin_dti            : num  2 3 11 14 13 11 5 10 24 14 ...
##  $ Default_flag       : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ No_of_Enquiry      : num  0 1 1 1 0 0 3 0 1 2 ...
##  $ enq_buckets        : chr  "0" "1-4" "1-4" "1-4" ...
##  $ annual_inc         : num  110000 135000 75000 51000 41500 ...
##  $ Income_bins        : num  9 11 6 4 3 4 12 7 6 4 ...
##  $ home_ownership     : chr  "MORTGAGE" "RENT" "MORTGAGE" "RENT" ...
##  $ purpose            : chr  "credit_card" "other" "educational" "credit_card" ...
##  $ open_acc           : num  6 3 7 5 8 5 4 7 6 9 ...
##  $ emp_length         : chr  "LT 1year" "10+ years" "2 years" "1 year" ...
##  $ verification_status: chr  "Not Verified" "Source Verified" "Source Verified" "Source Verified" ..
##  $ delinq_2yrs        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ loan_amnt          : num  7000 2000 12000 9350 6000 ...
##  $ Bins_loan_amt      : num  6 2 10 8 5 8 5 10 2 8 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   member_id = col_character(),
##   ..   loan_status = col_character(),
##   ..   int_rate = col_character(),
##   ..   Bin_int = col_double(),
##   ..   dti = col_double(),
##   ..   Bin_dti = col_double(),
##   ..   Default_flag = col_double(),
##   ..   No_of_Enquiry = col_double(),
##   ..   enq_buckets = col_character(),
##   ..   annual_inc = col_double(),
##   ..   Income_bins = col_double(),
##   ..   home_ownership = col_character(),
##   ..   purpose = col_character(),
##   ..   open_acc = col_double(),
##   ..   emp_length = col_character(),
```

```
##   ..   verification_status = col_character(),
##   ..   delinq_2yrs = col_double(),
##   ..   loan_amnt = col_double(),
##   ..   Bins_loan_amt = col_double()
##   .. )
##  - attr(*, ".internal.selfref")=<externalptr>
```

Data Cleaning

We can see by the str() function that the int_rate has a '%' symbol which will hinder our analysis further. We must clean that column. We also have many character data type columns which need to either by ordinal or nominal factors

```
Lend[, member_id := factor(member_id)]
Lend[, loan_status := factor(loan_status)]
Lend[, home_ownership:= factor(home_ownership)]
Lend[, purpose := factor(purpose)]
Lend[, verification_status := factor(verification_status)]

Lend[, int_rate := gsub('[%]', '', int_rate)]
Lend[, int_rate := trimws(int_rate)]
Lend[, int_rate := suppressWarnings(as.numeric(int_rate))]

Lend[open_acc %in% c(1,2,3,4,5), 'x' := 'LT5']
Lend[open_acc %in% c(6,7,8,9,10), 'x' := '6-10']
Lend[open_acc %in% c(11,12,13,14,15), 'x' := '11-15']
Lend[open_acc >15, 'x' := '15+']
Lend = Lend %>%
  rename(no_of_acct = x)
str(Lend)
```

```
## Classes 'data.table' and 'data.frame':   35808 obs. of  20 variables:
##  $ member_id          : Factor w/ 35808 levels "LC1","LC10","LC100",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ loan_status        : Factor w/ 2 levels "Charged Off",..: 1 2 2 2 2 2 2 2 2 2 ...
##  $ int_rate           : num  11.7 16 10.7 12.7 19.7 ...
##  $ Bin_int            : num  10 16 8 11 22 1 23 10 5 16 ...
##  $ dti                : num  1.06 2.61 11.34 14 13.01 ...
##  $ Bin_dti            : num  2 3 11 14 13 11 5 10 24 14 ...
##  $ Default_flag       : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ No_of_Enquiry      : num  0 1 1 1 0 0 3 0 1 2 ...
##  $ enq_buckets        : chr  "0" "1-4" "1-4" "1-4" ...
##  $ annual_inc         : num  110000 135000 75000 51000 41500 ...
##  $ Income_bins        : num  9 11 6 4 3 4 12 7 6 4 ...
##  $ home_ownership     : Factor w/ 5 levels "MORTGAGE","NONE",..: 1 5 1 5 1 1 1 5 5 1 ...
##  $ purpose            : Factor w/ 14 levels "car","credit_card",..: 2 10 4 2 3 3 8 2 10 3 ...
##  $ open_acc           : num  6 3 7 5 8 5 4 7 6 9 ...
##  $ emp_length         : chr  "LT 1year" "10+ years" "2 years" "1 year" ...
##  $ verification_status: Factor w/ 3 levels "Not Verified",..: 1 2 2 2 3 3 1 1 1 2 ...
##  $ delinq_2yrs        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ loan_amnt          : num  7000 2000 12000 9350 6000 ...
##  $ Bins_loan_amt      : num  6 2 10 8 5 8 5 10 2 8 ...
##  $ no_of_acct         : chr  "6-10" "LT5" "6-10" "LT5" ...
##  - attr(*, "spec")=
```

```
##   .. cols(
##   ..   member_id = col_character(),
##   ..   loan_status = col_character(),
##   ..   int_rate = col_character(),
##   ..   Bin_int = col_double(),
##   ..   dti = col_double(),
##   ..   Bin_dti = col_double(),
##   ..   Default_flag = col_double(),
##   ..   No_of_Enquiry = col_double(),
##   ..   enq_buckets = col_character(),
##   ..   annual_inc = col_double(),
##   ..   Income_bins = col_double(),
##   ..   home_ownership = col_character(),
##   ..   purpose = col_character(),
##   ..   open_acc = col_double(),
##   ..   emp_length = col_character(),
##   ..   verification_status = col_character(),
##   ..   delinq_2yrs = col_double(),
##   ..   loan_amnt = col_double(),
##   ..   Bins_loan_amt = col_double()
##   .. )
##  - attr(*, ".internal.selfref")=<externalptr>
##  - attr(*, "index")= int
##   ..- attr(*, "__open_acc")= int  75 113 157 195 377 382 458 611 628 642 ...
```

Data Splitting

We must split out dataset into diffirent training testing datasets for further analysis. We also split our data into Defaulters and Non Defaulters.

```
#Training Testing

## 75% of the sample size
smp_size = floor(0.75 * nrow(Lend))

## set the seed to make our partition reproducible
set.seed(123)
train_ind = sample(seq_len(nrow(Lend)), size = smp_size)

train = Lend[train_ind, ]
test = Lend[-train_ind, ]

#default & nondefault data

defaultdata =  filter(Lend, Default_flag == 1)
nondefault = filter(Lend, Default_flag == 0)
view(defaultdata)
view(nondefault)
defaultdata = setDT(defaultdata)
nondefault = setDT(nondefault)
```

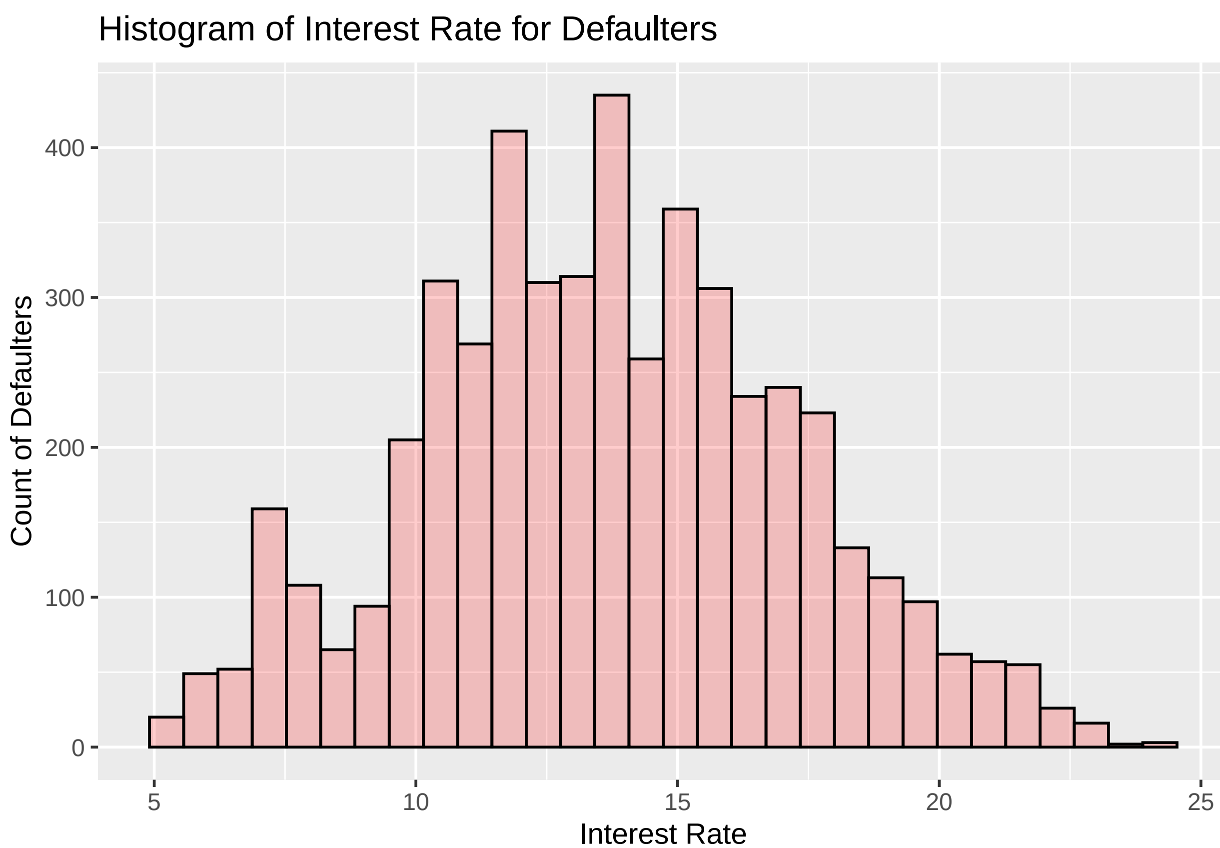
Data Exploration

Let us start by exploring the density of defaulters and nondefaulters accros the total density for each variables, i.e int__rate, dti, loan__amnt, annual__inc

Let us first see the histogram count of defaulters on interest rates:

```
ggplot(data = defaultdata, aes(x = int_rate)) +
  geom_histogram(color = "black", fill = "red", alpha = 0.2) +
  labs(title = 'Histogram of Interest Rate for Defaulters',
       y = 'Count of Defaulters', x = 'Interest Rate')
```

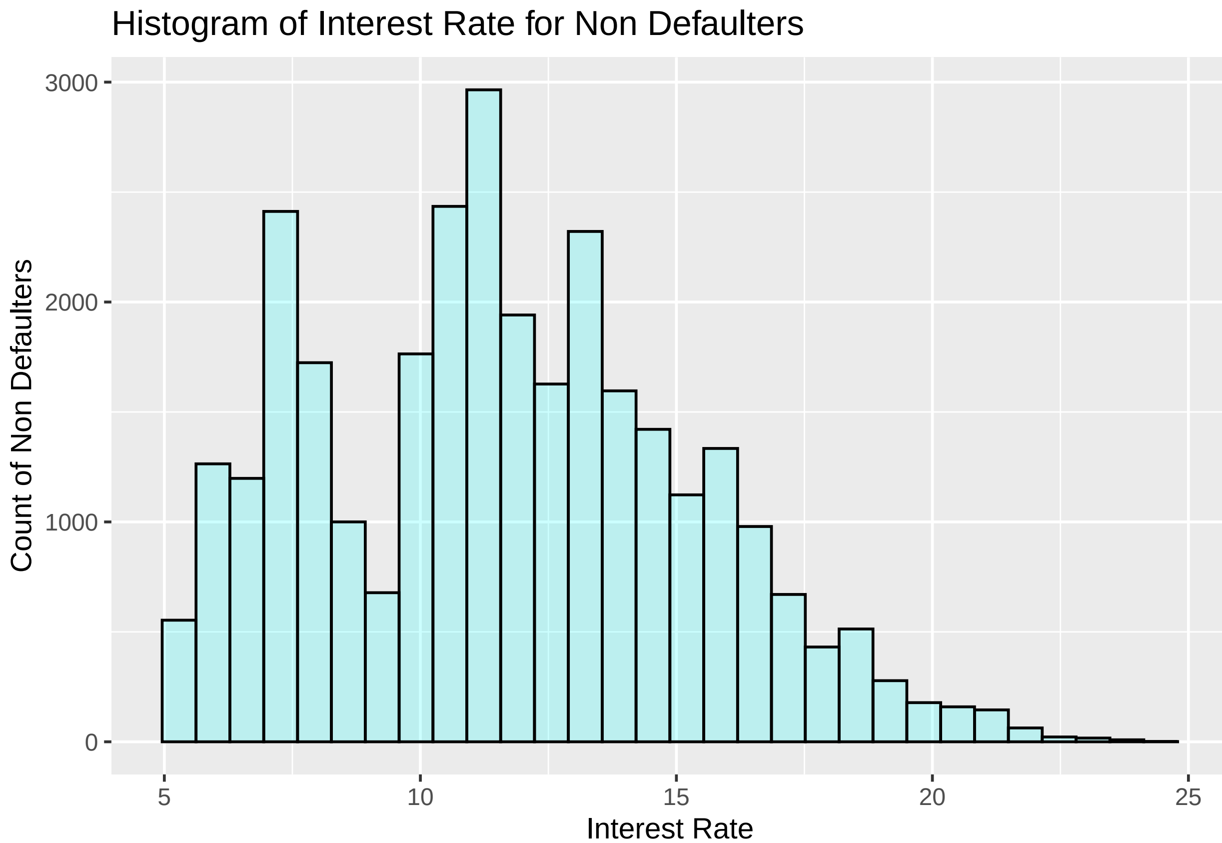
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Let us see the histogram count of non defaulters on interest rates:

```
ggplot(data = nondefault, aes(x = int_rate)) +
  geom_histogram(color = "black", fill = "#00FFFF", alpha = 0.2) +
  labs(title = 'Histogram of Interest Rate for Non Defaulters',
       y = 'Count of Non Defaulters', x = 'Interest Rate' )
```

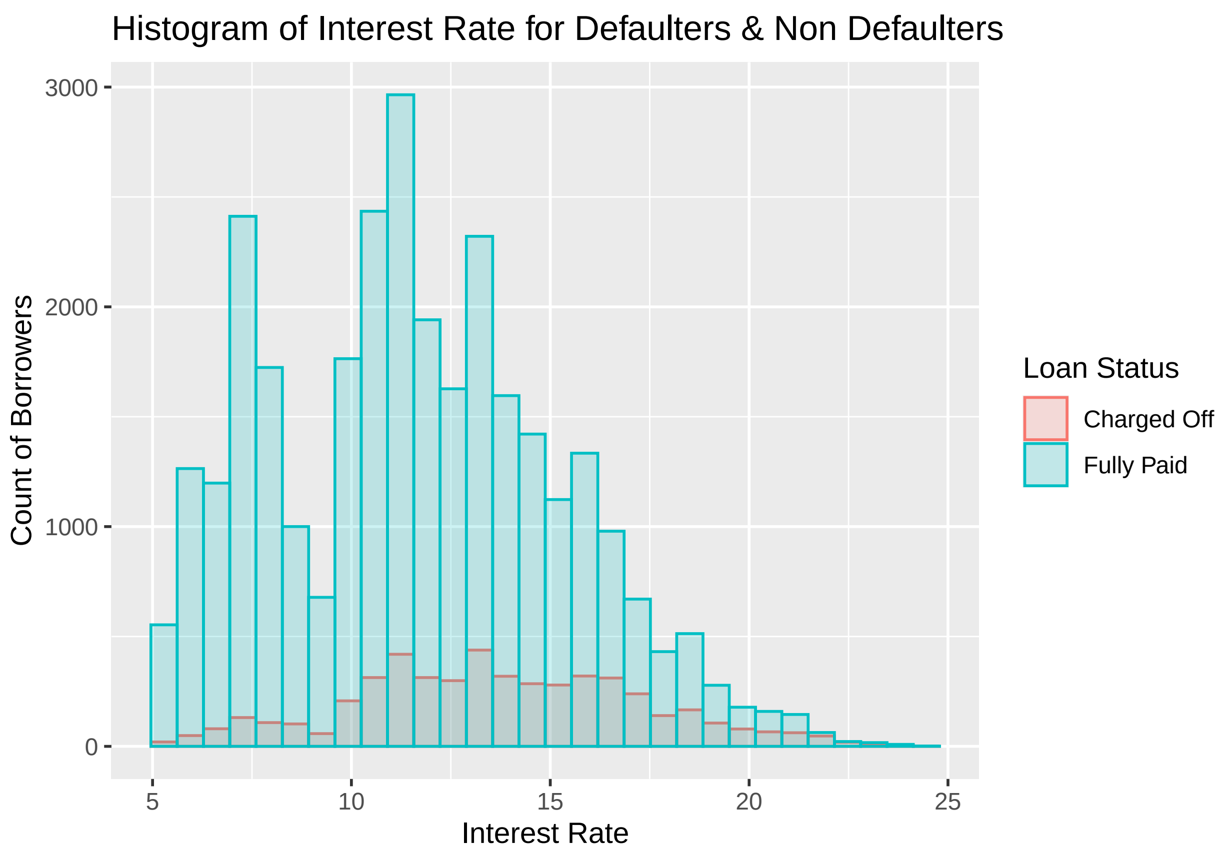
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Now let us compare the above histograms on one plot and a common scale against interest rates:

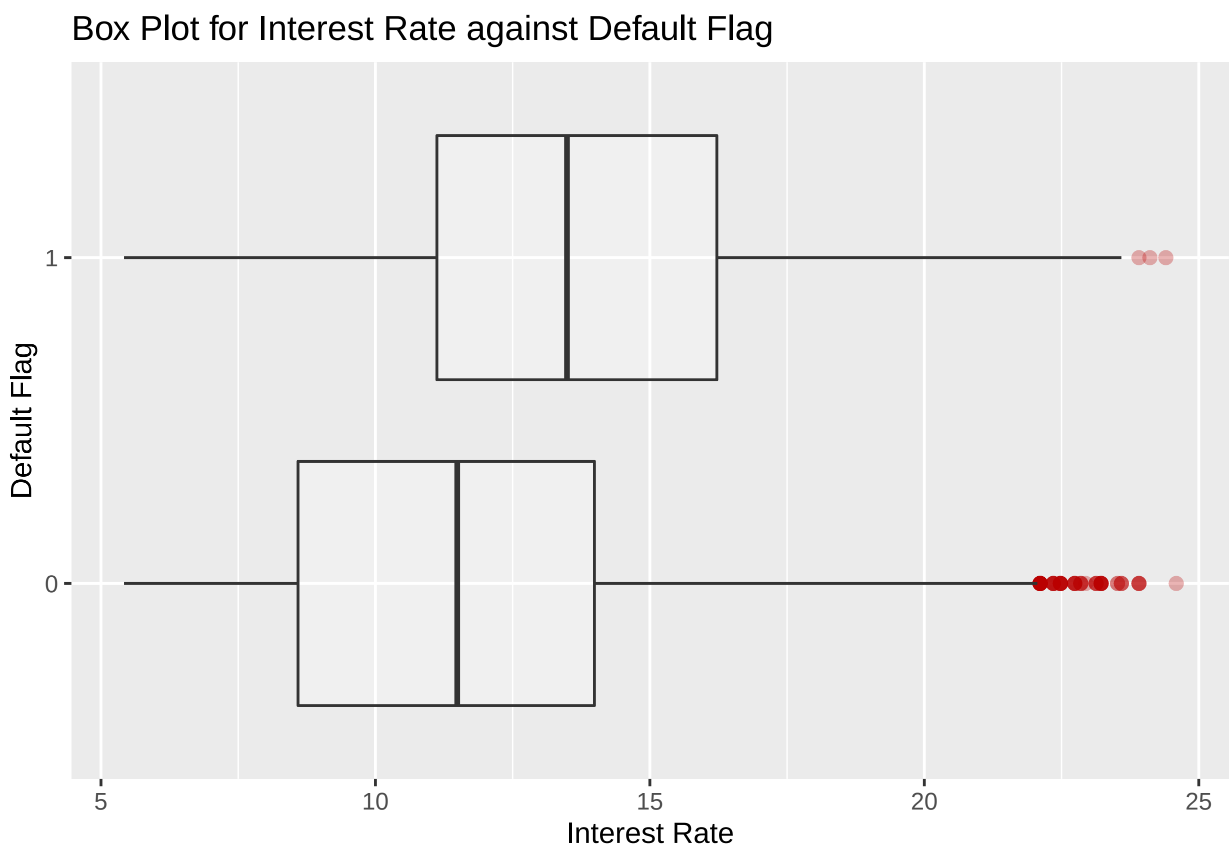
```
ggplot(Lend, aes(x = int_rate, color = loan_status, fill = loan_status)) +
  geom_histogram(alpha = 0.2, position = "identity") +
  labs(title = 'Histogram of Interest Rate for Defaulters & Non Defaulters',
       y = 'Count of Borrowers', x = 'Interest Rate',
       color = 'Loan Status', fill = 'Loan Status')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Let us plot a boxplot for interest rate against the default flag to see the outliers and the median:

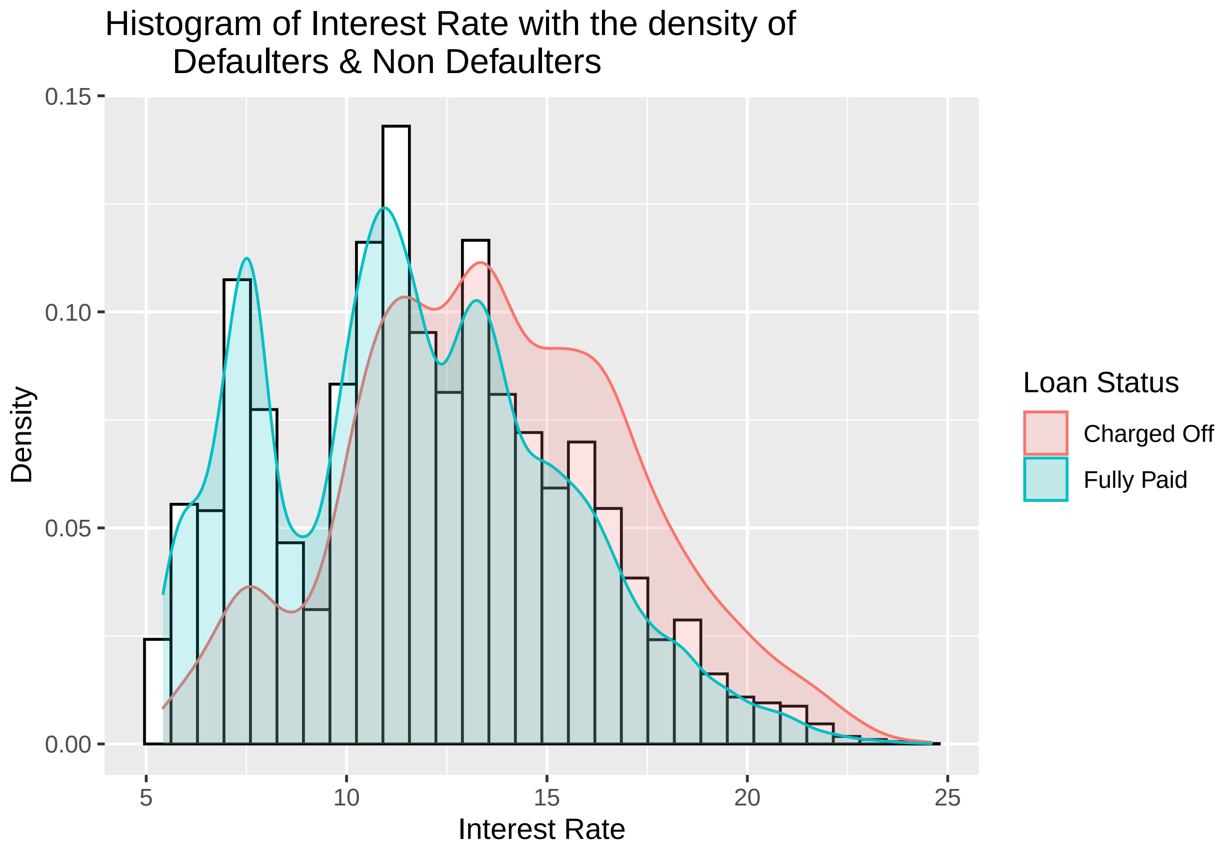
```
ggplot(data = Lend) +
  geom_boxplot(mapping = aes(x = reorder(Default_flag, int_rate, FUN = median),
                             y = int_rate),
               outlier.colour = "#BA0000", outlier.size = 2, alpha = 0.3) +
  labs(title = 'Box Plot for Interest Rate against Default Flag',
       y = 'Interest Rate', x = 'Default Flag') +
   coord_flip()
```



Let us observe the density of Defaulters and Non Defaulters on their interest rates, we might expect to get some insights:

```
ggplot(Lend, aes(x = int_rate, color = loan_status, fill = loan_status)) +
  geom_histogram(aes(y = ..density..), colour = "black", fill = "white") +
  geom_density(alpha = 0.2) +
  labs(title = 'Histogram of Interest Rate with the density of
       Defaulters & Non Defaulters', y = 'Density', x = 'Interest Rate',
       color = 'Loan Status', fill = 'Loan Status')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



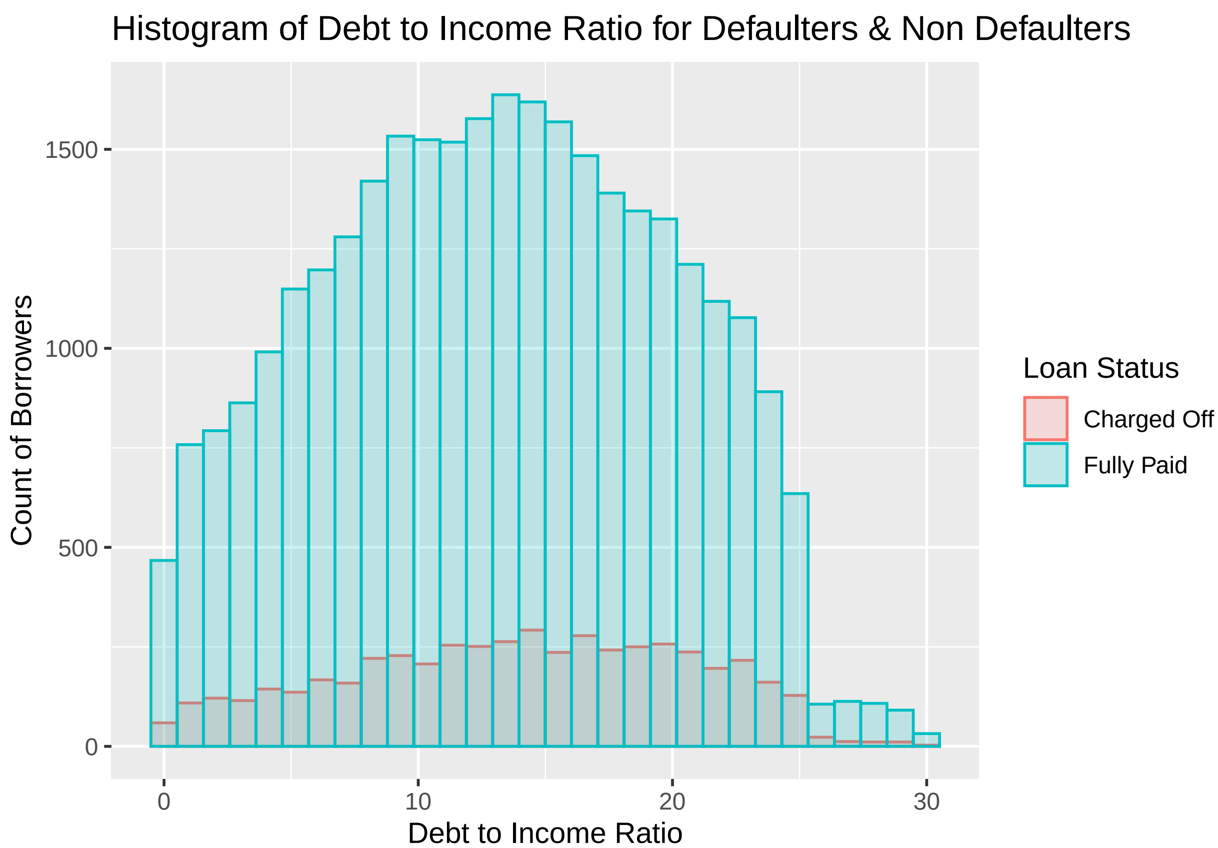
As we can see in the above plot that when the interest rates were low the default rates were proportionally low as compared to the non default rate. As we go on increasing the interest rate we can see that the density of defaulters increase while the density of non defaulters decrease. This can give us some good information about how the interest rate affects the default rates.

Now Let us perform the same plotting for the debt to income ratio i.e dti:

This is the Histogram overlap of Defaulters and Non Defaulters with their Debt to Income Ratio:

```
ggplot(Lend, aes(x = dti, color = loan_status, fill = loan_status)) +
  geom_histogram(alpha = 0.2, position = "identity") +
  labs(title = 'Histogram of Debt to Income Ratio for Defaulters & Non Defaulters',
       y = 'Count of Borrowers', x = 'Debt to Income Ratio',
       color = 'Loan Status', fill = 'Loan Status')
```

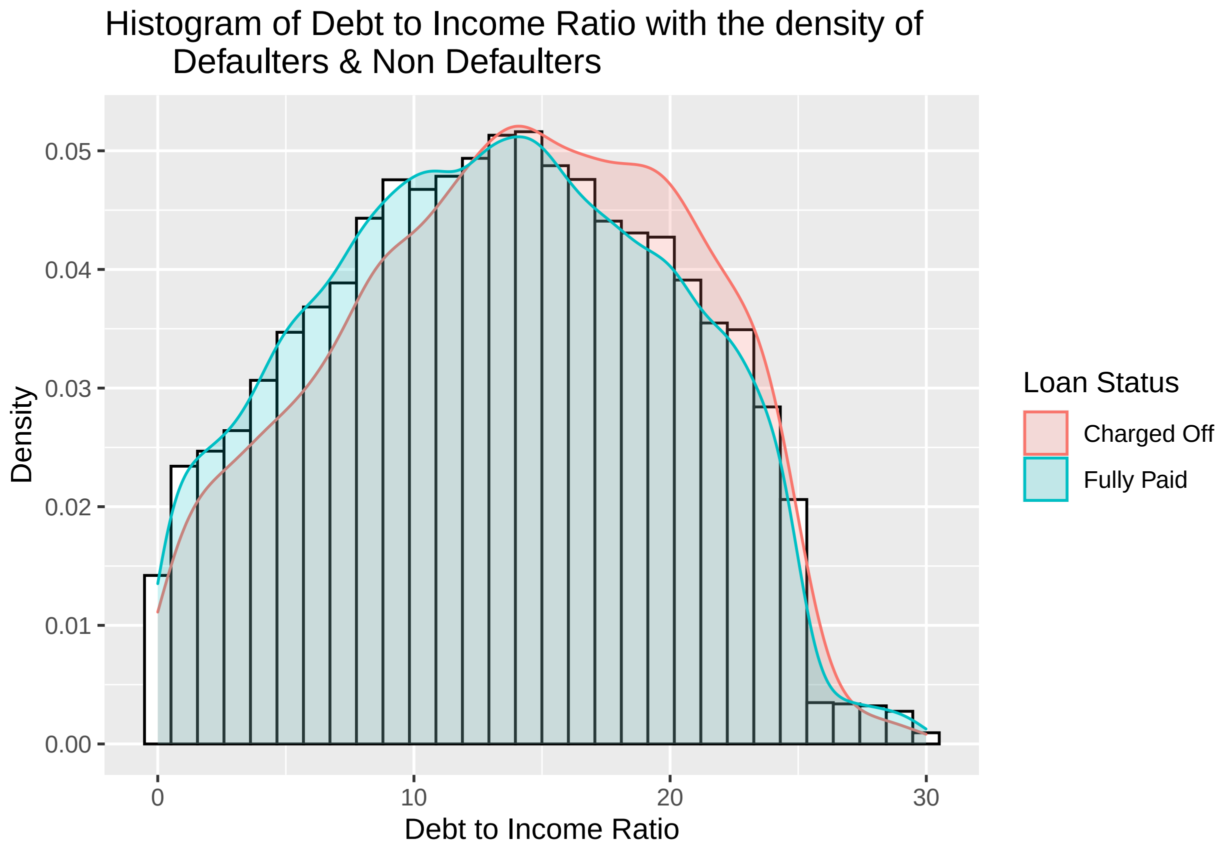
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Let us Plot the densities of each default flag values:

```
ggplot(Lend, aes(x = dti, color = loan_status, fill = loan_status)) +
  geom_histogram(aes(y = ..density..), colour = "black", fill = "white") +
  geom_density(alpha = 0.2) +
  labs(title = 'Histogram of Debt to Income Ratio with the density of
       Defaulters & Non Defaulters', y = 'Density', x = 'Debt to Income Ratio',
       color = 'Loan Status', fill = 'Loan Status')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

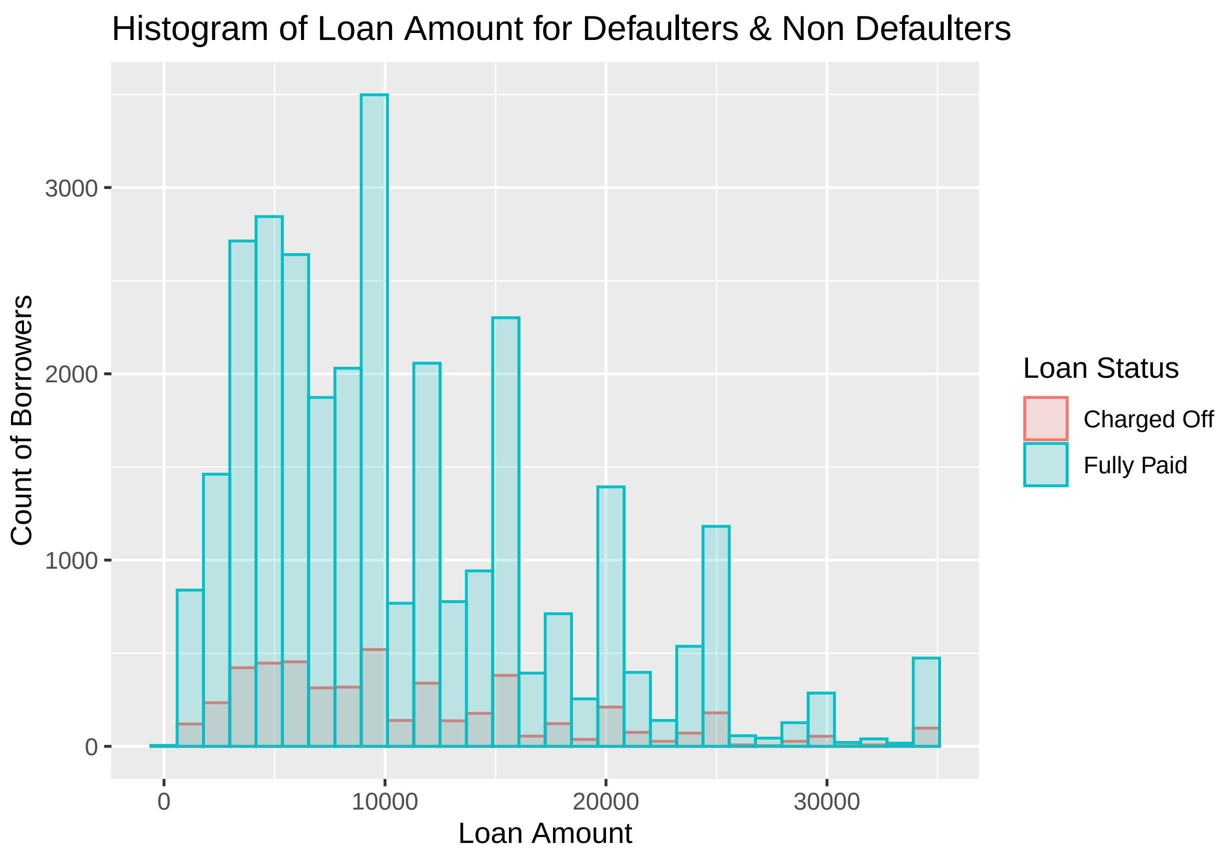


For the dti against loan_status we can observe a simmilar effect but for the dti greater than 25 we can see that the density of defaulters and non defaulters are almost equal.

Let us check the overlapping histogram of number of defaulters and non defaulters over their loan amount:

```
ggplot(Lend, aes(x = loan_amnt, color = loan_status, fill = loan_status)) +
  geom_histogram(alpha = 0.2, position = "identity") +
  labs(title = 'Histogram of Loan Amount for Defaulters & Non Defaulters',
       y = 'Count of Borrowers', x = 'Loan Amount',
       color = 'Loan Status', fill = 'Loan Status')
```

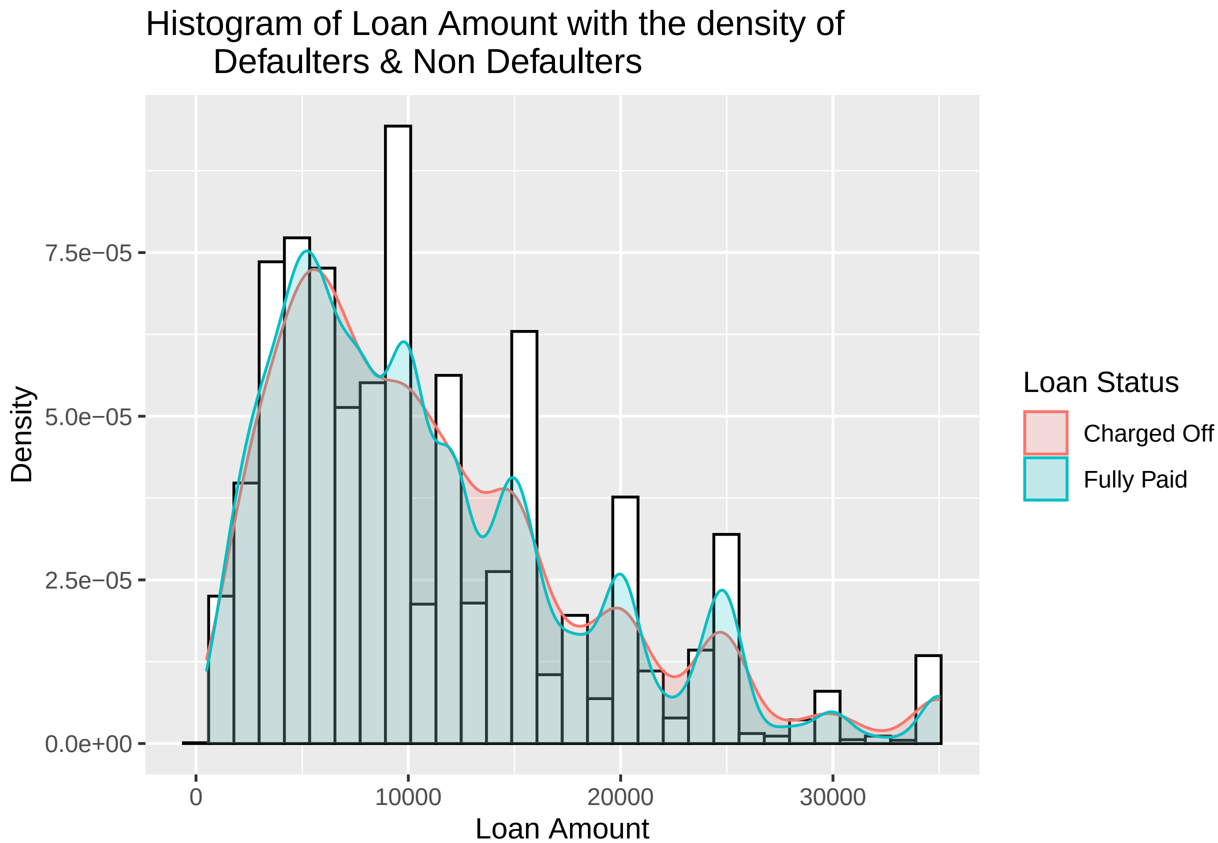
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Let us check the density against loan amounts:

```
ggplot(Lend, aes(x = loan_amnt, color = loan_status, fill = loan_status)) +
  geom_histogram(aes(y = ..density..), colour = "black", fill = "white") +
  geom_density(alpha = 0.2) +
  labs(title = 'Histogram of Loan Amount with the density of
       Defaulters & Non Defaulters', y = 'Density', x = 'Loan Amount',
       color = 'Loan Status', fill = 'Loan Status')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

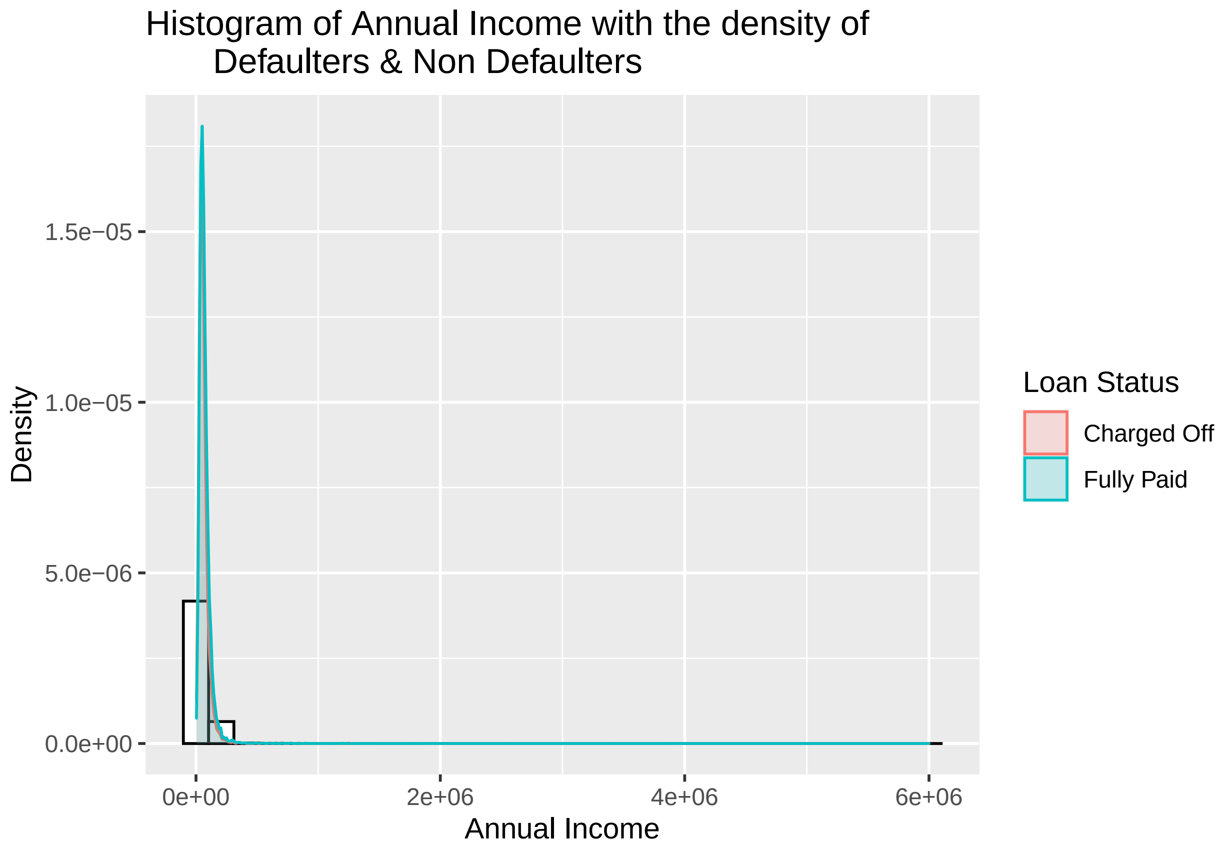


We dont have much information to get from this graph except that there are more borrowers borrowing loan near the 10,000 mark

Let us see for the annual income and the loan status:

```
ggplot(Lend, aes(x = annual_inc, color = loan_status, fill = loan_status)) +
  geom_histogram(aes(y = ..density..), colour = "black", fill = "white") +
  geom_density(alpha = 0.2) +
  labs(title = 'Histogram of Annual Income with the density of
       Defaulters & Non Defaulters', y = 'Density', x = 'Annual Income',
       color = 'Loan Status', fill = 'Loan Status')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



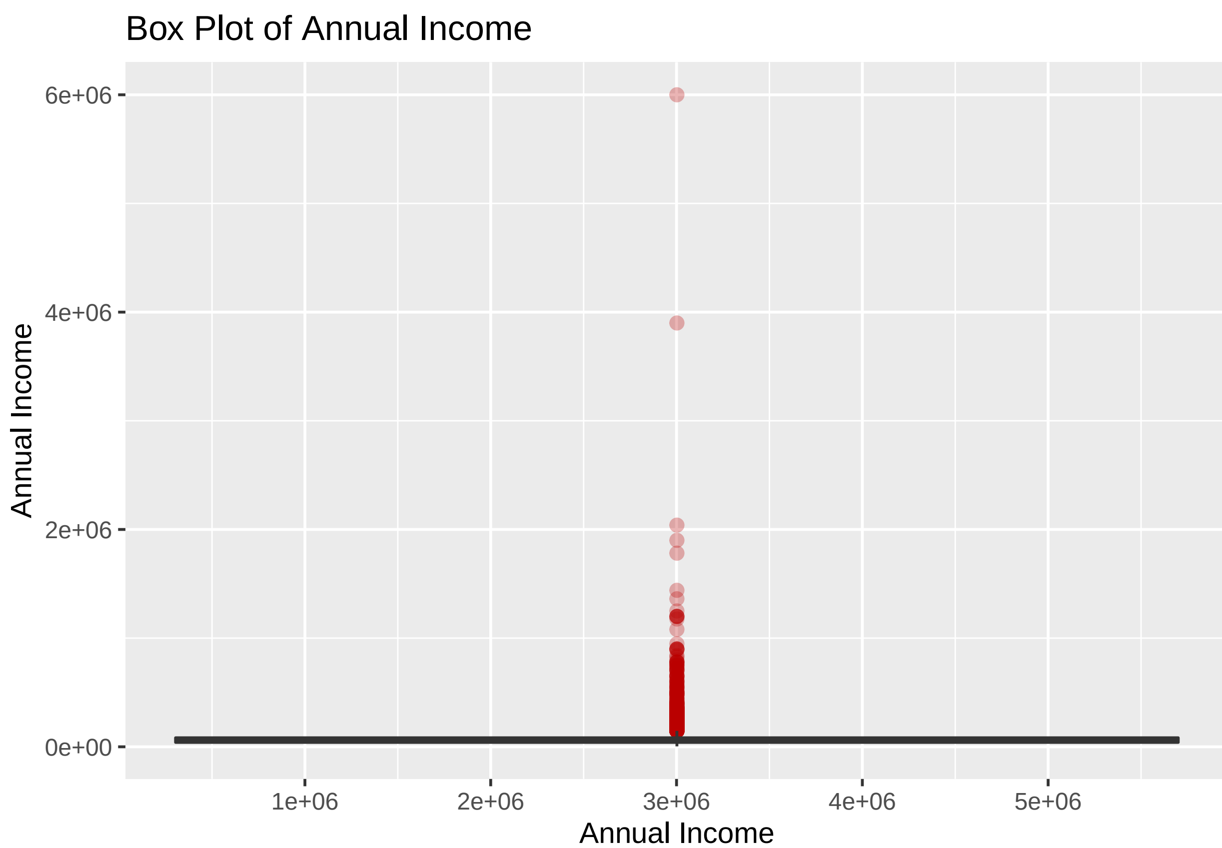
This graph looks very weird. This might be due to outliers in this column.

Let us check for the summary and outliers:

```
summary(Lend$annual_inc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    4000   40500   59449   69278   83000 6000000
```

```
ggplot(Lend, aes(x = annual_inc, y = annual_inc, group = 1)) +
  geom_boxplot(outlier.colour = "#BA0000", outlier.size = 2, alpha = 0.3) +
  labs(title = 'Box Plot of Annual Income', y = 'Annual Income', x = 'Annual Income')
```

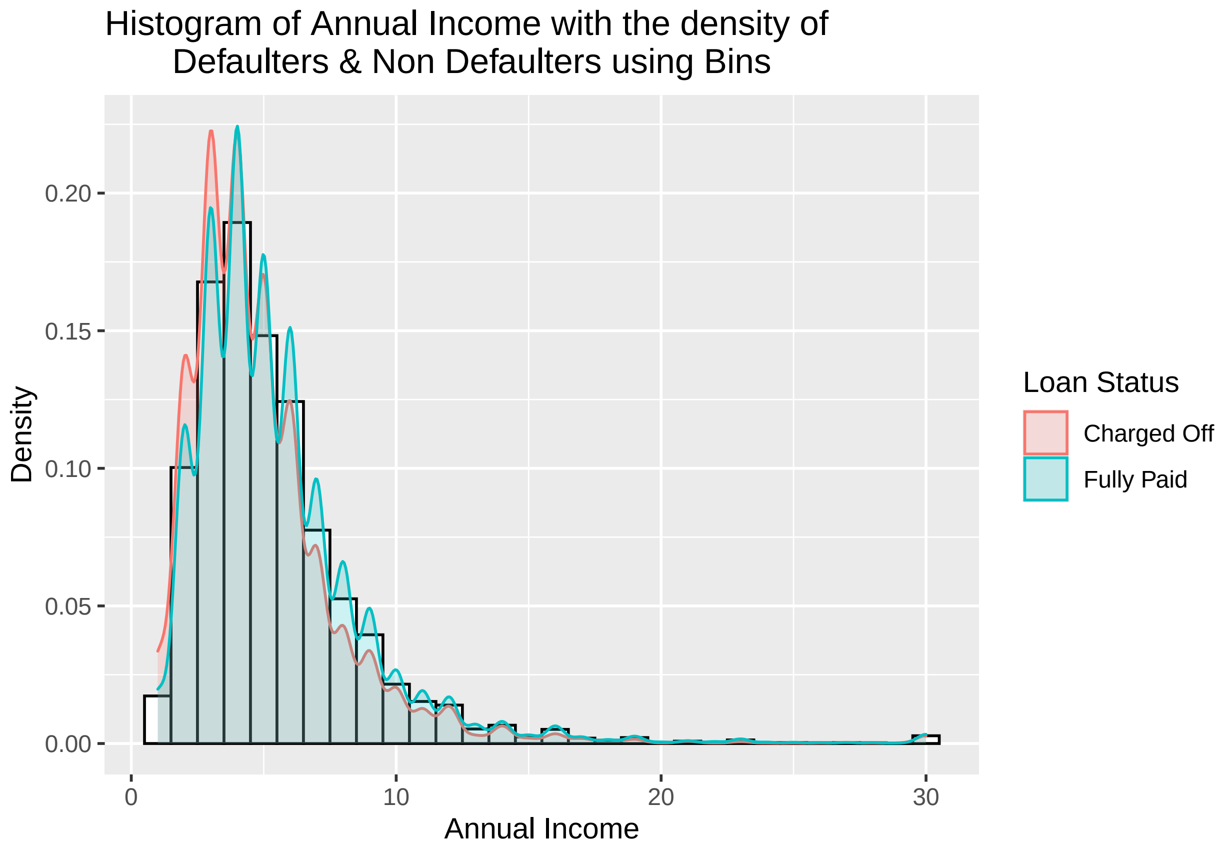


We can see that the max value for the annual income is 6,000,000 and the 3rd Quartile ends at 83000 which is a clear mark that this is a big outlier problem. We can see in the plot as well that there are many outliers.

Let us use the Income_bins for controlling the outlier problem:

```
ggplot(Lend, aes(x = Income_bins, color = loan_status, fill = loan_status)) +
  geom_histogram(aes(y = ..density..), colour = "black", fill = "white") +
  geom_density(alpha = 0.2) +
  labs(title = 'Histogram of Annual Income with the density of
       Defaulters & Non Defaulters using Bins', y = 'Density', x = 'Annual Income',
       color = 'Loan Status', fill = 'Loan Status')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Now we can see that there is a slight difference in the density of defauters. there are more defaulters when the annual income is gets low

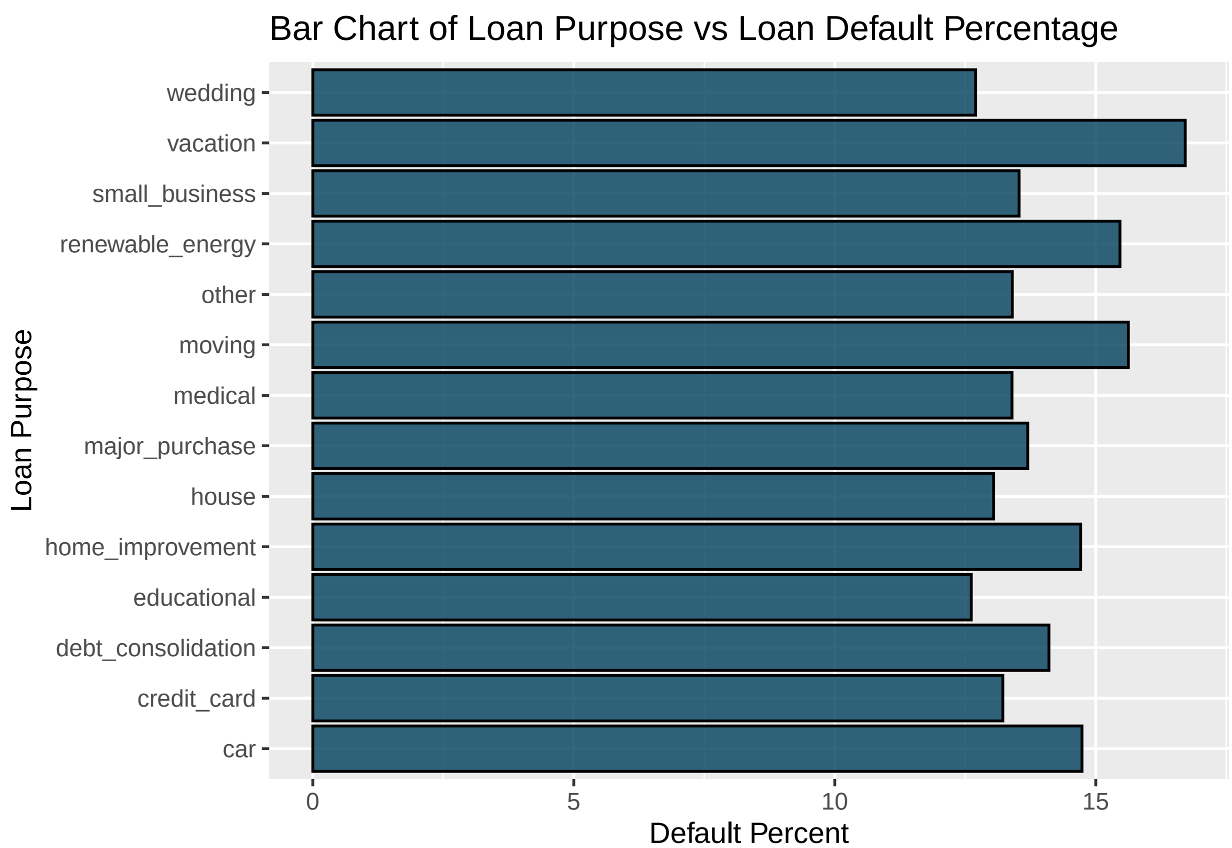
Let us explore the percentages of defaulters accros every unique value of purpose, open account bins, No of enquiry, delinquent im last 2 years

Let us start with plotting perecntage of defaulters for each loan purpose:

```
purpose = Lend %>%
  group_by(purpose) %>%
  summarize(cnt = n(), defcnt = sum(Default_flag)) %>%
  summarize(purpose, percent = (defcnt*100)/cnt)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
ggplot(purpose, aes(x = percent, y = purpose)) +
  geom_bar(stat = 'identity', fill = '#003F5C', color = 'black', alpha = 0.8) +
  labs(title = 'Bar Chart of Loan Purpose vs Loan Default Percentage ',
       y = 'Loan Purpose', x = 'Default Percent')
```



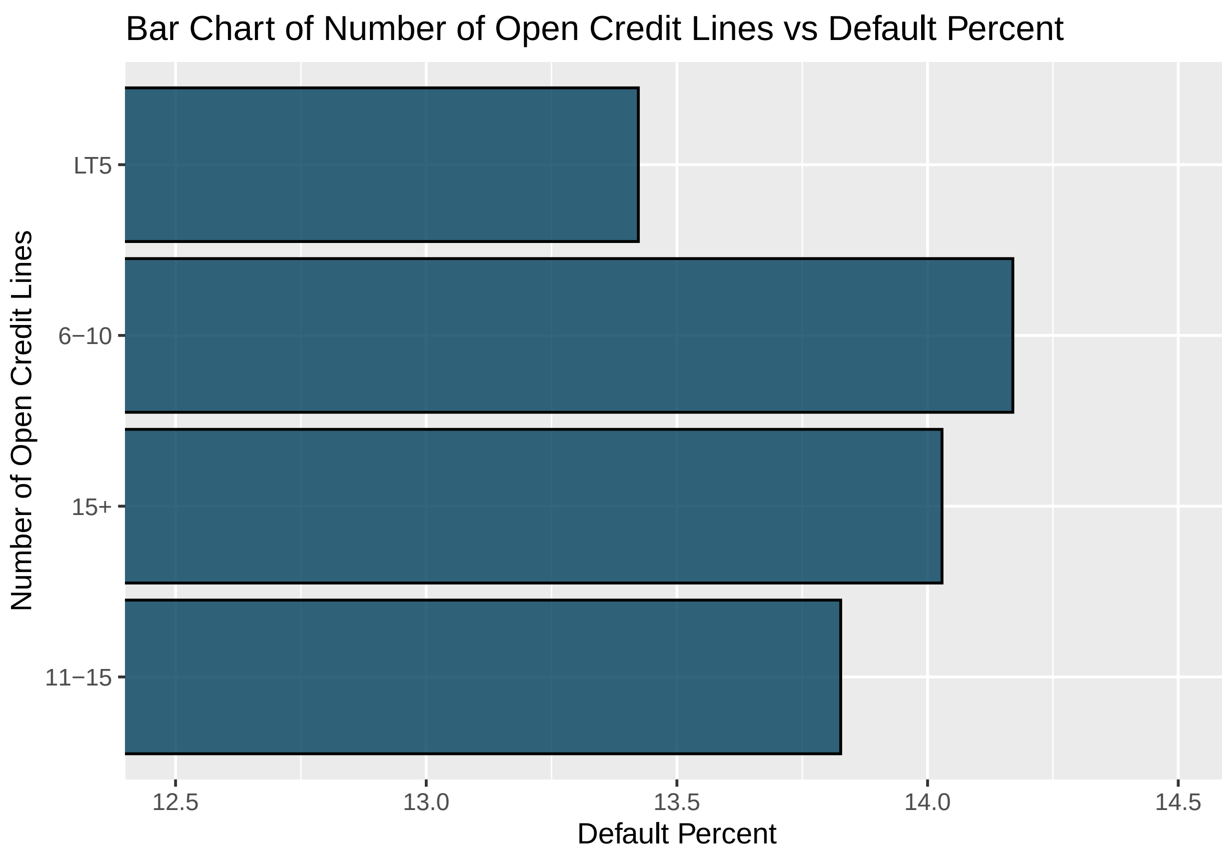
We can point out that loans taken for vacations have higher defaulter proportions and educational, house, and wedding have less defaulter proportions

Let us plot another graph where we can check the percentage of defaulters over each distinct number of open credit lines:

```
acctvsdefault = Lend %>%
  group_by(no_of_acct) %>%
  summarize(cnt = n(), defcnt = sum(Default_flag)) %>%
  summarize(no_of_acct, percent = (defcnt*100)/cnt)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
ggplot(acctvsdefault, aes(x = percent,y = no_of_acct)) +
  geom_bar(stat = 'identity',fill = '#003F5C', color = 'black', alpha = 0.8) +
  coord_cartesian(xlim = c(12.5,14.5)) +
  labs(title = 'Bar Chart of Number of Open Credit Lines vs Default Percent',
       y = 'Number of Open Credit Lines', x = 'Default Percent')
```



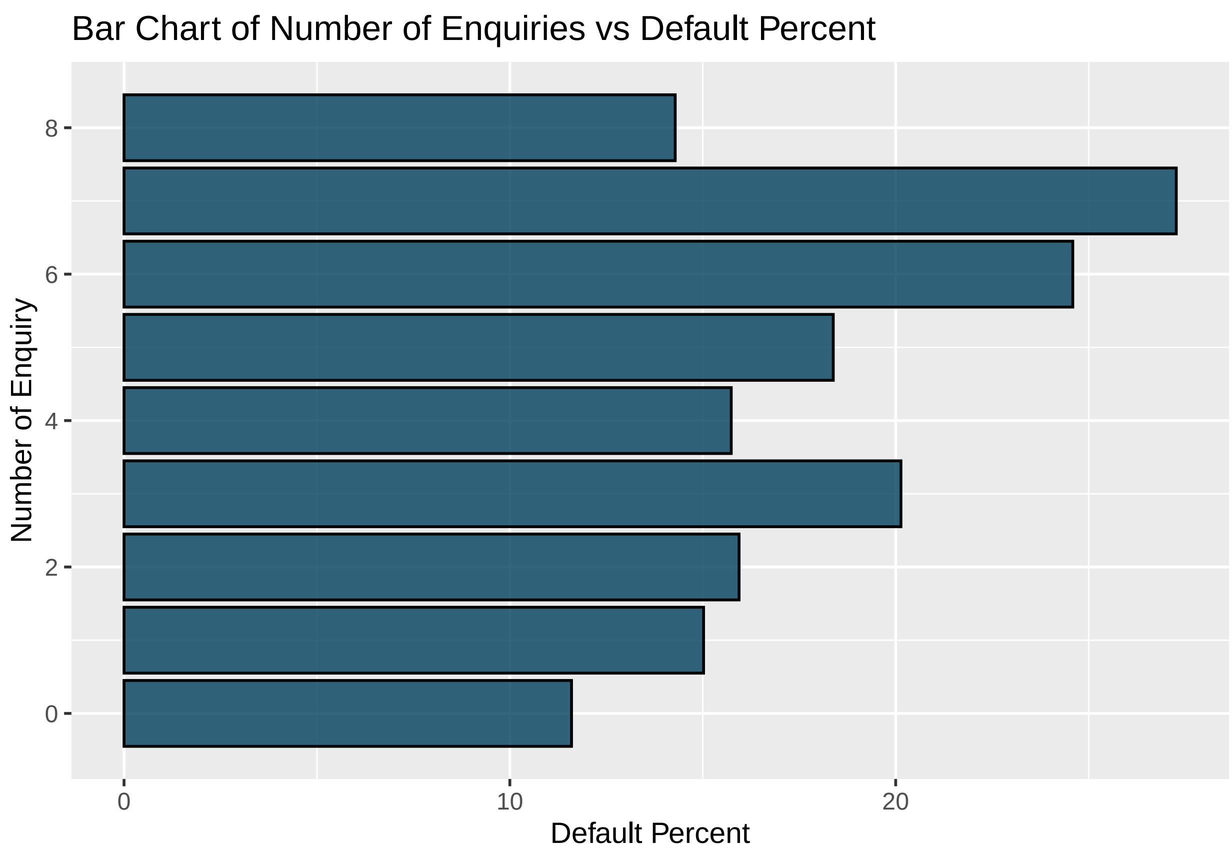
We can obsserve that borrowers with less than 5 credit lines have less default percentage whereas the borrowers having credit lines between 6-10 have the highest and the ones having more than 15 is yet less.

Let us plot a graph for percentage of defaults for every number of enquiry the borrower made in the last 6 months:

```
inq = Lend %>%
  group_by(No_of_Enquiry) %>%
  summarize(cnt = n(), defcnt = sum(Default_flag)) %>%
  summarize(No_of_Enquiry, percent = (defcnt*100)/cnt)
```

```
## ‘summarise()‘ ungrouping output (override with ‘.groups‘ argument)
```

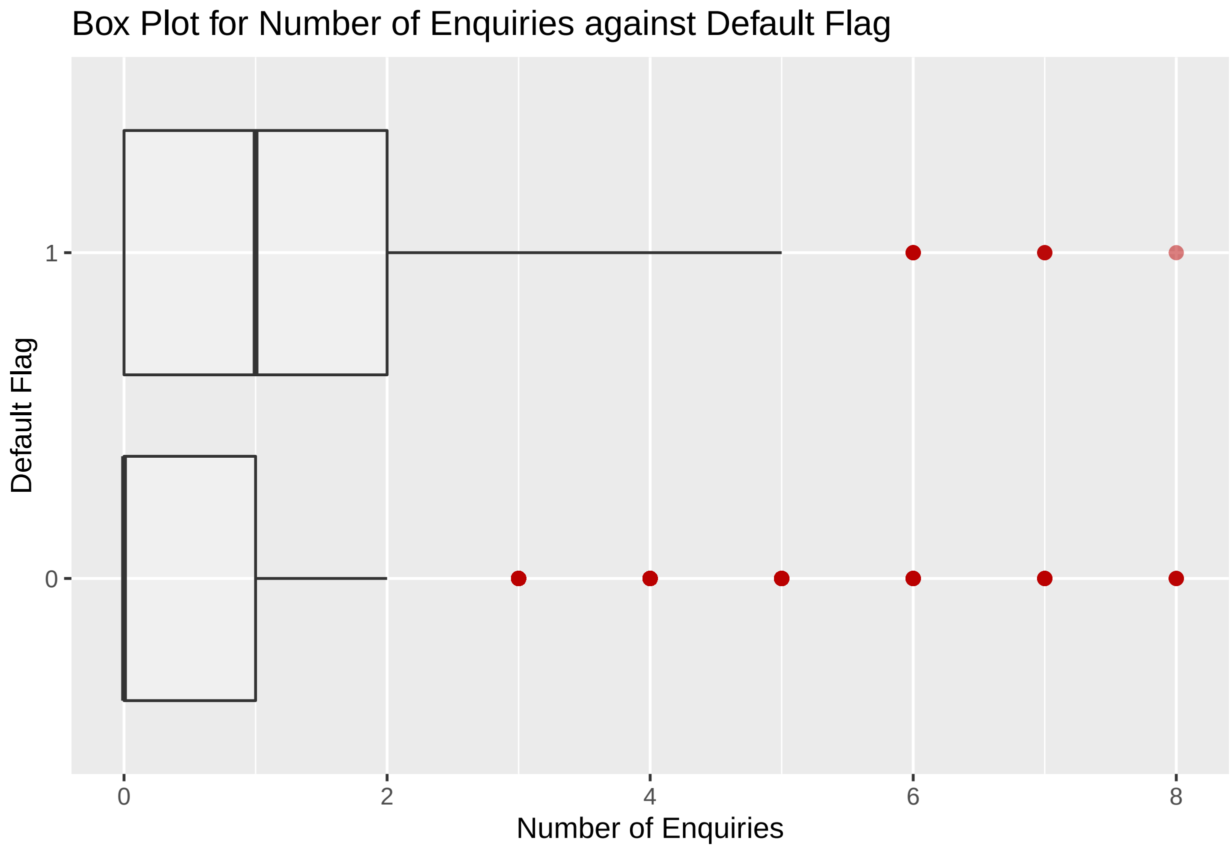
```
ggplot(inq) +
  geom_bar( mapping = aes(y = percent, x = No_of_Enquiry), stat = 'identity',
            fill = '#003F5C', color = 'black', alpha = 0.8) +
  coord_flip() +
  labs(title = 'Bar Chart of Number of Enquiries vs Default Percent',
       y = 'Default Percent', x = 'Number of Enquiry')
```



We observe that as the number of enquiries increase the percentage of default rates increase but thats not the case for 8 enquiries in the last 6 months. We need to further explore to understand why is it so

Let us plot a boxplot for Number of Enquiries against the default flag:

```
ggplot(data = Lend) +
  geom_boxplot(mapping = aes(x = reorder(Default_flag, No_of_Enquiry, FUN = median),
                             y=No_of_Enquiry,),
               outlier.colour = "#BA0000", outlier.size = 2, alpha = 0.3) +
  labs(title = 'Box Plot for Number of Enquiries against Default Flag',
       y = 'Number of Enquiries', x = 'Default Flag') +
  coord_flip()
```

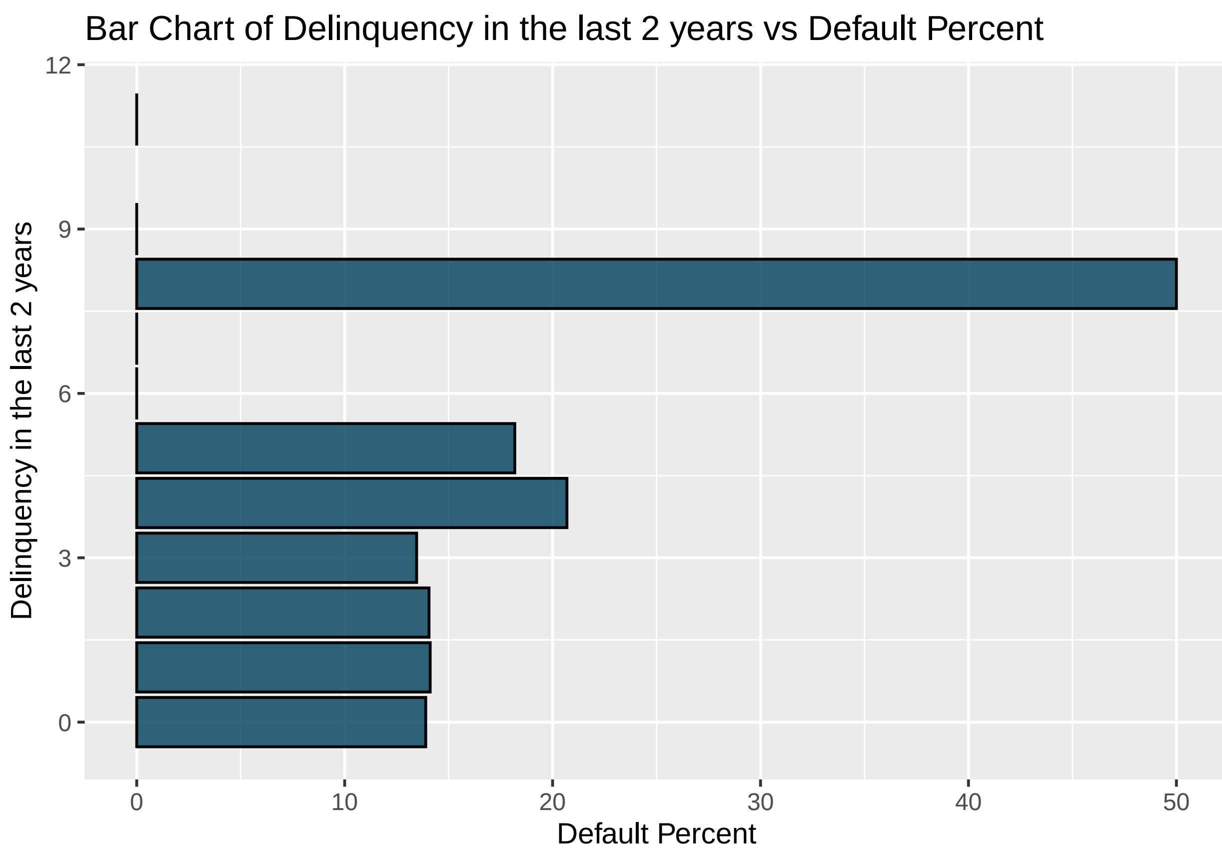


Let us also explore how are the percentages of defaults related to past 2 year delinqueny:

```
delinq = Lend %>%
  group_by(delinq_2yrs) %>%
  summarize(cnt = n(), defcnt = sum(Default_flag)) %>%
  summarize(delinq_2yrs, percent = (defcnt*100)/cnt)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

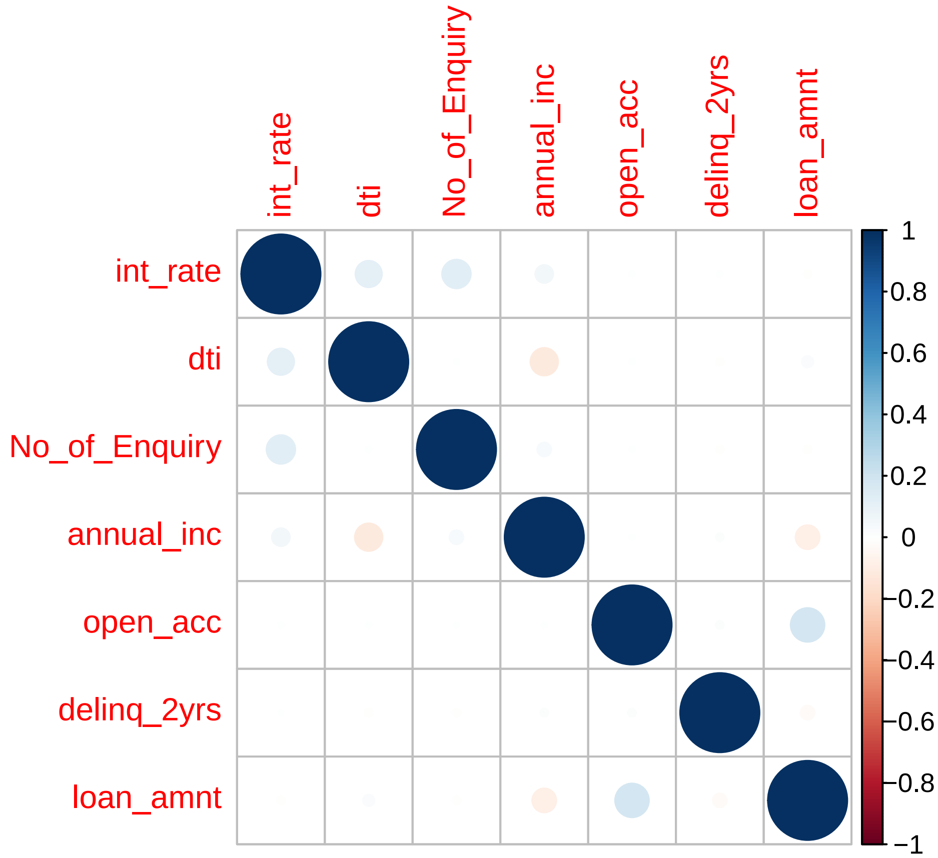
```
ggplot(delinq) +
  geom_bar( mapping = aes(y = percent, x = delinq_2yrs), stat='identity',
            fill = '#003F5C', color = 'black', alpha = 0.8) +
  coord_flip() +
  labs(title = 'Bar Chart of Delinquency in the last 2 years vs Default Percent',
       y = 'Default Percent', x = 'Delinquency in the last 2 years')
```



We can observe that as delinquency increase from 0 to 3 there is'nt any significant increase in default percentage but at delinq_2 = 8 the default percent jumps to 50%.

Let us check the correlation of each variable by pairs

```
mat <- round(cor(Lend[, c(3,5,8,10,14,17,18)], use = "pair"), 2)
corrplot(mat)
```



We can see the correlation of each variable to each other. Annual income looks neggatively correlated to the debt to income ratio. Loan amount also looks negatively correlated to the anual income and have a slightly hiher positive correlation with the number of open credit lines.

We can use this information we gathered in our EDA for further analysis where we do our hypothesis testing