Capstone

Suprasanna Pradhan

1 September 2019

1. Objective of the project This project aims at the case of customers' default payments in Taiwan. Expected to calculate the probability of default for a customer and further use multiple models to compare their predictive accuracy. From the perspective of risk management, the result of predictive accuracy of the estimated probability of default will be more valuable than the binary result of classification - credible or not credible clients.
2. Introduction A Taiwan-based bank wants to better predict the likelihood of default for its customers, as well as identify the key drivers that determine this likelihood. This would inform the issuer's decisions on who to give a credit card to and what credit limit to provide. It would also help the bank have a better understanding of their current and potential customers, which would inform their future strategy, including their planning of offering targeted credit products to their customers. Our fundamental objective is to help the bank to improve its credit card services for the mutual benefit of customers and the business itself. We will try to touch on the most popular methods and algorithms in order to find the best model which will help predict default and to answer the questions:
3. Check probability of default payment via categories of different demographic variables?
4. Variables are having strong implications to default payment?

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(caretEnsemble)
```

```
##
## Attaching package: 'caretEnsemble'
```

```
## The following object is masked from 'package:ggplot2':
##
##     autoplot
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(Amelia)
```

```
## Loading required package: Rcpp
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.5, built: 2018-05-07)
## ## Copyright (C) 2005-2019 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
library(mice)
```

```
##
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:tidyr':
##
##     complete
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
##
## Attaching package: 'GGally'
```

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

```
library(gutenbergr)
library(tidytext)
library(dplyr)
library(janeaustenr)
library(stringi)
library(tidyr)
library(rpart)
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:psych':
##
##     outlier
```

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

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

This dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005

```
library(readxl)
#library(XLConnect)
#Importing Data set
setwd("C:/Users/SuprasannaPradhan/Documents/My Files/Great Lakes Projects/Capstone Pro
ject TCD")
taiwan_data=read_excel("Taiwan-Customer defaults.xls", skip=1)
str(taiwan_data)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    30000 obs. of  25 variables:
##  $ ID                        : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ LIMIT_BAL                 : num  20000 120000 90000 50000 50000 50000 500000 100
000 140000 20000 ...
##  $ SEX                       : num  2 2 2 2 1 1 1 2 2 1 ...
##  $ EDUCATION                 : num  2 2 2 2 2 1 1 2 3 3 ...
##  $ MARRIAGE                  : num  1 2 2 1 1 2 2 2 1 2 ...
##  $ AGE                       : num  24 26 34 37 57 37 29 23 28 35 ...
##  $ PAY_0                     : num  2 -1 0 0 -1 0 0 0 0 -2 ...
##  $ PAY_2                     : num  2 2 0 0 0 0 0 -1 0 -2 ...
##  $ PAY_3                     : num  -1 0 0 0 -1 0 0 -1 2 -2 ...
##  $ PAY_4                     : num  -1 0 0 0 0 0 0 0 0 -2 ...
##  $ PAY_5                     : num  -2 0 0 0 0 0 0 0 0 -1 ...
##  $ PAY_6                     : num  -2 2 0 0 0 0 0 -1 0 -1 ...
##  $ BILL_AMT1                 : num  3913 2682 29239 46990 8617 ...
##  $ BILL_AMT2                 : num  3102 1725 14027 48233 5670 ...
##  $ BILL_AMT3                 : num  689 2682 13559 49291 35835 ...
##  $ BILL_AMT4                 : num  0 3272 14331 28314 20940 ...
##  $ BILL_AMT5                 : num  0 3455 14948 28959 19146 ...
##  $ BILL_AMT6                 : num  0 3261 15549 29547 19131 ...
##  $ PAY_AMT1                  : num  0 0 1518 2000 2000 ...
##  $ PAY_AMT2                  : num  689 1000 1500 2019 36681 ...
##  $ PAY_AMT3                  : num  0 1000 1000 1200 10000 657 38000 0 432 0 ...
##  $ PAY_AMT4                  : num  0 1000 1000 1100 9000 ...
##  $ PAY_AMT5                  : num  0 0 1000 1069 689 ...
##  $ PAY_AMT6                  : num  0 2000 5000 1000 679 ...
##  $ default payment next month: num  1 1 0 0 0 0 0 0 0 0 ...
```

We got here total 30000 observations with 25 variables , considering that the data set has been classified in 4 categories as follow : Variable X1 to X5 is general data where Limit balance and age are continuous frequencies and other variable are bionomical categories . Variable X6 to X11 is payment history from April to September2005 which is consisted of categorical variables Variable X12 to X17 is bill statement amount ,all are carrying amount values X18 to X23 is carried out the previous amount paid by the customer shows in frequency Default will be the dependent variable which we should be used for predict our probabilities

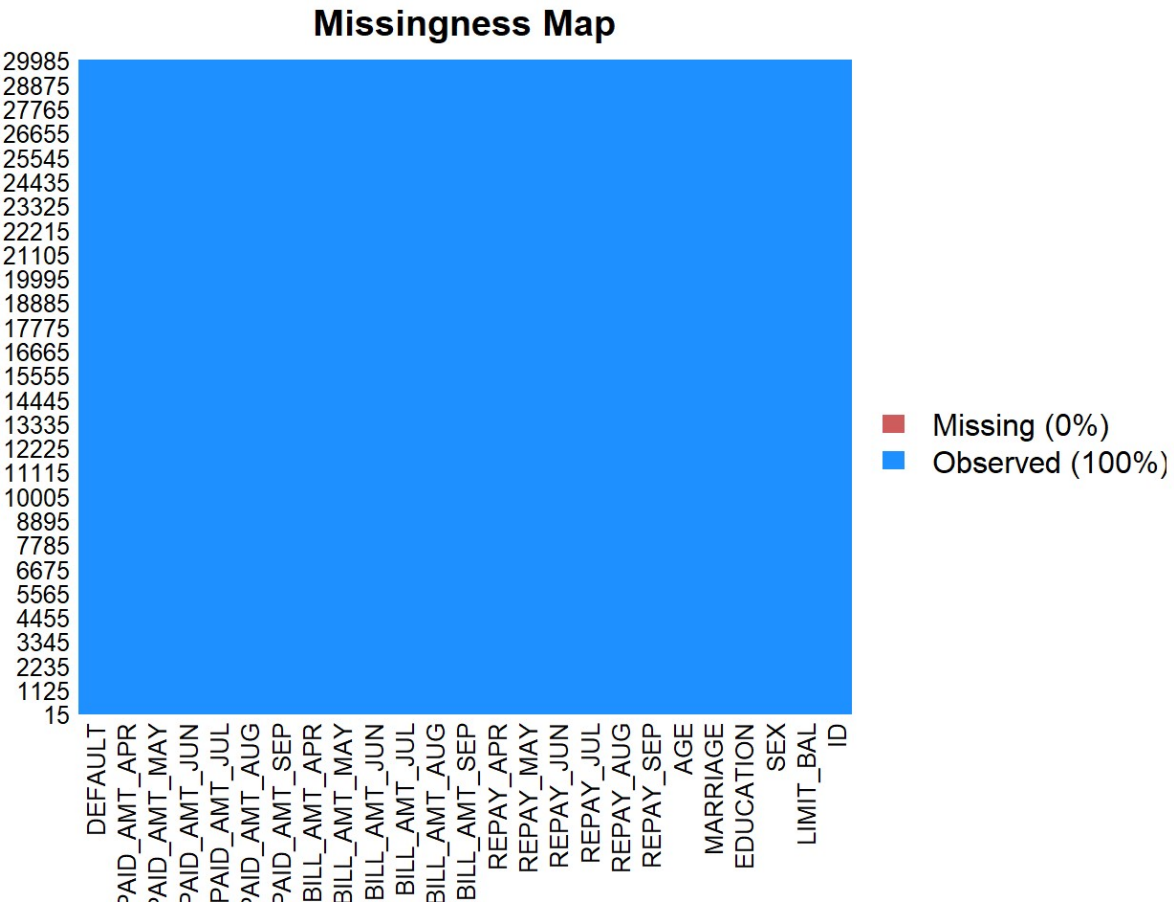
```
#Changing varibale names
colnames(taiwan_data)[colnames(taiwan_data)=="PAY_0"] <- "REPAY_SEP"
colnames(taiwan_data)[colnames(taiwan_data)=="PAY_2"] <- "REPAY_AUG"
colnames(taiwan_data)[colnames(taiwan_data)=="PAY_3"] <- "REPAY_JUL"
colnames(taiwan_data)[colnames(taiwan_data)=="PAY_4"] <- "REPAY_JUN"
colnames(taiwan_data)[colnames(taiwan_data)=="PAY_5"] <- "REPAY_MAY"
colnames(taiwan_data)[colnames(taiwan_data)=="PAY_6"] <- "REPAY_APR"
colnames(taiwan_data)[colnames(taiwan_data)=="BILL_AMT1"] <- "BILL_AMT_SEP"
colnames(taiwan_data)[colnames(taiwan_data)=="BILL_AMT2"] <- "BILL_AMT_AUG"
colnames(taiwan_data)[colnames(taiwan_data)=="BILL_AMT3"] <- "BILL_AMT_JUL"
colnames(taiwan_data)[colnames(taiwan_data)=="BILL_AMT4"] <- "BILL_AMT_JUN"
colnames(taiwan_data)[colnames(taiwan_data)=="BILL_AMT5"] <- "BILL_AMT_MAY"
colnames(taiwan_data)[colnames(taiwan_data)=="BILL_AMT6"] <- "BILL_AMT_APR"
colnames(taiwan_data)[colnames(taiwan_data)=="PAY_AMT1"] <- "PAID_AMT_SEP"
colnames(taiwan_data)[colnames(taiwan_data)=="PAY_AMT2"] <- "PAID_AMT_AUG"
colnames(taiwan_data)[colnames(taiwan_data)=="PAY_AMT3"] <- "PAID_AMT_JUL"
colnames(taiwan_data)[colnames(taiwan_data)=="PAY_AMT4"] <- "PAID_AMT_JUN"
colnames(taiwan_data)[colnames(taiwan_data)=="PAY_AMT5"] <- "PAID_AMT_MAY"
colnames(taiwan_data)[colnames(taiwan_data)=="PAY_AMT6"] <- "PAID_AMT_APR"
colnames(taiwan_data)[colnames(taiwan_data)=="default payment next month"] <- "DEFAUL
T"
taiwan_bank <-as.data.frame(taiwan_data)
```

Above we have renamed all variables

```
##Checking Data ##
names(taiwan_bank)
```

```
##  [1] "ID"           "LIMIT_BAL"    "SEX"          "EDUCATION"
##  [5] "MARRIAGE"     "AGE"          "REPAY_SEP"    "REPAY_AUG"
##  [9] "REPAY_JUL"    "REPAY_JUN"    "REPAY_MAY"    "REPAY_APR"
## [13] "BILL_AMT_SEP" "BILL_AMT_AUG" "BILL_AMT_JUL" "BILL_AMT_JUN"
## [17] "BILL_AMT_MAY" "BILL_AMT_APR" "PAID_AMT_SEP" "PAID_AMT_AUG"
## [21] "PAID_AMT_JUL" "PAID_AMT_JUN" "PAID_AMT_MAY" "PAID_AMT_APR"
## [25] "DEFAULT"
```

```
#visualize the missing data
missmap(taiwan_bank)
```




```
sum(is.na(taiwan_bank))
```

```
## [1] 0
```

```
taiwan_bank[is.na(taiwan_bank)] <- 0
sum(is.na(taiwan_bank))
```

```
## [1] 0
```

There is no missing data , hence we proceed further and converted some variables level into categorical data as we need them for visualization purpose. The variables like SEX, MARRAIGE, EDUCATION levels are converted into categorical data

```
# Renaming levels of variables
library(plyr)
```

```
## ------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following object is masked from 'package:purrr':
##
##     compact
```

```
taiwan_bank$ID <- as.factor(taiwan_bank$ID )
taiwan_bank$SEX <- as.factor(taiwan_bank$SEX )
taiwan_bank$EDUCATION  <- as.factor(taiwan_bank$EDUCATION)
taiwan_bank$MARRIAGE <- as.factor(taiwan_bank$MARRIAGE )
taiwan_bank$DEFAULT <- as.factor(taiwan_bank$DEFAULT )
levels(taiwan_bank$DEFAULT) <- c("No","Yes")
levels(taiwan_bank$SEX)[levels(taiwan_bank$SEX) == "1"] <- "Male"
levels(taiwan_bank$SEX)[levels(taiwan_bank$SEX) == "2"] <- "Female"
levels(taiwan_bank$EDUCATION)[levels(taiwan_bank$EDUCATION) == "1"] <- "Graduated"
levels(taiwan_bank$EDUCATION)[levels(taiwan_bank$EDUCATION) == "2"] <- "University"
levels(taiwan_bank$EDUCATION)[levels(taiwan_bank$EDUCATION) == "3"] <- "High.School"
levels(taiwan_bank$EDUCATION)[levels(taiwan_bank$EDUCATION) =="4"] <- "Others"
levels(taiwan_bank$EDUCATION)[levels(taiwan_bank$EDUCATION) =="5"] <- "Unlisted"
levels(taiwan_bank$EDUCATION)[levels(taiwan_bank$EDUCATION) =="6"] <- "Unlisted_one"
levels(taiwan_bank$MARRIAGE)[levels(taiwan_bank$MARRIAGE) == "1"] <- "Married"
levels(taiwan_bank$MARRIAGE)[levels(taiwan_bank$MARRIAGE) == "2"] <- "Unmarried"
levels(taiwan_bank$MARRIAGE)[levels(taiwan_bank$MARRIAGE) == "3"] <- "Others"
write.csv(taiwan_bank, file = "taiwan_bank.xls")
str(taiwan_bank)
```

```
## 'data.frame':    30000 obs. of  25 variables:
##  $ ID          : Factor w/ 30000 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
...
##  $ LIMIT_BAL   : num  20000 120000 90000 50000 50000 50000 500000 100000 140000 200
00 ...
##  $ SEX         : Factor w/ 2 levels "Male","Female": 2 2 2 2 1 1 1 2 2 1 ...
##  $ EDUCATION   : Factor w/ 7 levels "0","Graduated",..: 3 3 3 3 3 2 2 3 4 4 ...
##  $ MARRIAGE    : Factor w/ 4 levels "0","Married",..: 2 3 3 2 2 3 3 3 2 3 ...
##  $ AGE         : num  24 26 34 37 57 37 29 23 28 35 ...
##  $ REPAY_SEP   : num  2 -1 0 0 -1 0 0 0 0 -2 ...
##  $ REPAY_AUG   : num  2 2 0 0 0 0 0 -1 0 -2 ...
##  $ REPAY_JUL   : num  -1 0 0 0 -1 0 0 -1 2 -2 ...
##  $ REPAY_JUN   : num  -1 0 0 0 0 0 0 0 0 -2 ...
##  $ REPAY_MAY   : num  -2 0 0 0 0 0 0 0 0 -1 ...
##  $ REPAY_APR   : num  -2 2 0 0 0 0 0 -1 0 -1 ...
##  $ BILL_AMT_SEP: num  3913 2682 29239 46990 8617 ...
##  $ BILL_AMT_AUG: num  3102 1725 14027 48233 5670 ...
##  $ BILL_AMT_JUL: num  689 2682 13559 49291 35835 ...
##  $ BILL_AMT_JUN: num  0 3272 14331 28314 20940 ...
##  $ BILL_AMT_MAY: num  0 3455 14948 28959 19146 ...
##  $ BILL_AMT_APR: num  0 3261 15549 29547 19131 ...
##  $ PAID_AMT_SEP: num  0 0 1518 2000 2000 ...
##  $ PAID_AMT_AUG: num  689 1000 1500 2019 36681 ...
##  $ PAID_AMT_JUL: num  0 1000 1000 1200 10000 657 38000 0 432 0 ...
##  $ PAID_AMT_JUN: num  0 1000 1000 1100 9000 ...
##  $ PAID_AMT_MAY: num  0 0 1000 1069 689 ...
##  $ PAID_AMT_APR: num  0 2000 5000 1000 679 ...
##  $ DEFAULT     : Factor w/ 2 levels "No","Yes": 2 2 1 1 1 1 1 1 1 1 ...
```

```
attach(taiwan_bank)
dim(taiwan_bank)
```

```
## [1] 30000    25
```

```
head(taiwan_bank)
```

```
##   ID LIMIT_BAL    SEX  EDUCATION  MARRIAGE AGE REPAY_SEP REPAY_AUG
## 1  1     20000 Female University   Married  24         2         2
## 2  2    120000 Female University Unmarried  26        -1         2
## 3  3     90000 Female University Unmarried  34         0         0
## 4  4     50000 Female University   Married  37         0         0
## 5  5     50000   Male University   Married  57        -1         0
## 6  6     50000   Male  Graduated Unmarried  37         0         0
##   REPAY_JUL REPAY_JUN REPAY_MAY REPAY_APR BILL_AMT_SEP BILL_AMT_AUG
## 1        -1        -1        -2        -2         3913         3102
## 2         0         0         0         2         2682         1725
## 3         0         0         0         0        29239        14027
## 4         0         0         0         0        46990        48233
## 5        -1         0         0         0         8617         5670
## 6         0         0         0         0        64400        57069
##   BILL_AMT_JUL BILL_AMT_JUN BILL_AMT_MAY BILL_AMT_APR PAID_AMT_SEP
## 1          689            0            0            0            0
## 2         2682         3272         3455         3261            0
## 3        13559        14331        14948        15549         1518
## 4        49291        28314        28959        29547         2000
## 5        35835        20940        19146        19131         2000
## 6        57608        19394        19619        20024         2500
##   PAID_AMT_AUG PAID_AMT_JUL PAID_AMT_JUN PAID_AMT_MAY PAID_AMT_APR DEFAULT
## 1          689            0            0            0            0     Yes
## 2         1000         1000         1000            0         2000     Yes
## 3         1500         1000         1000         1000         5000      No
## 4         2019         1200         1100         1069         1000      No
## 5        36681        10000         9000          689          679      No
## 6         1815          657         1000         1000          800      No
```

Precentage of defaulter

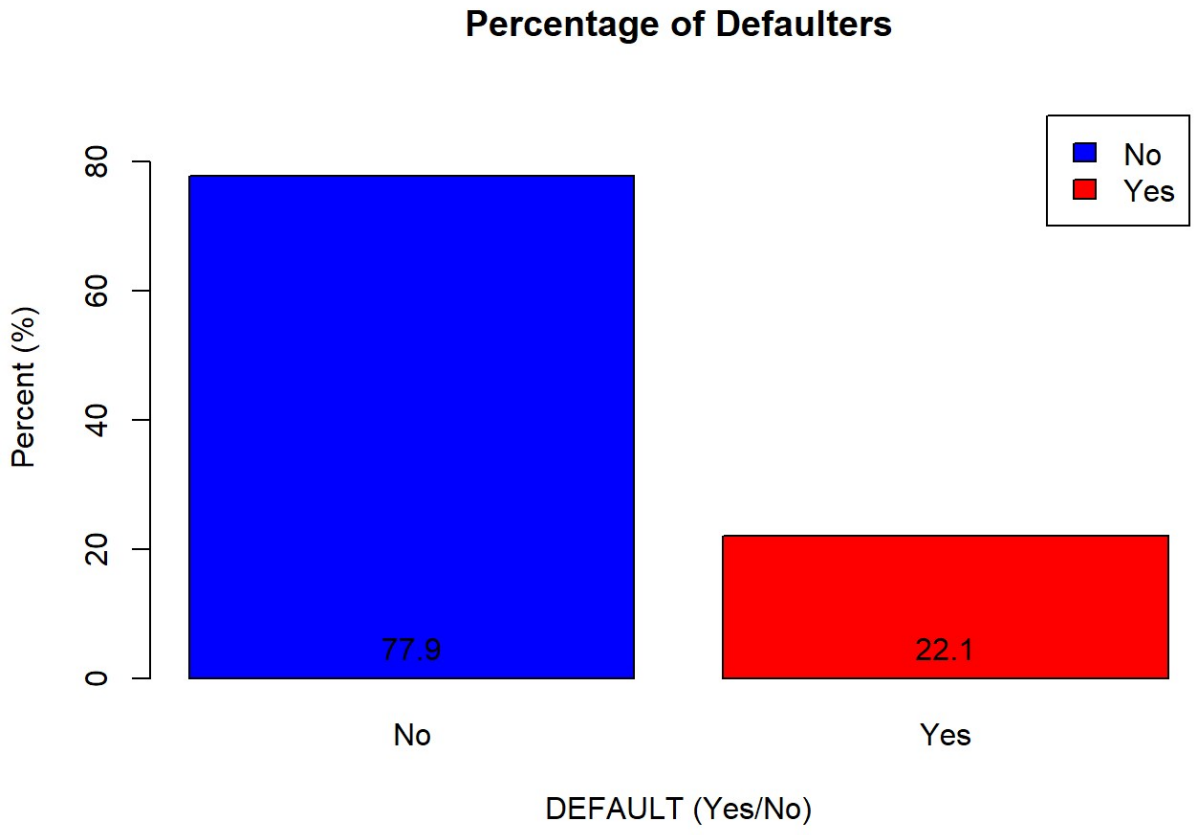
```
# printing counts of defaulters
per_defaults<- table(DEFAULT)
# proportion of defaulters
defaulters <- prop.table(per_defaults)
# Percentages of defaulters
round(defaulters*100,2)
```

```
## DEFAULT
##    No   Yes
## 77.88 22.12
```

```
nrow(subset(taiwan_bank, DEFAULT == "Yes"))/nrow(taiwan_data)
```

```
## [1] 0.2212
```

```
# percentage of defaulters
library(psych)
#describe(taiwan_data)[, c(1:5, 8:9)]
tab1 <- round(prop.table(table(taiwan_bank$DEFAULT))*100,2)
# bar-plot
bp <- barplot(tab1,
        xlab = "DEFAULT (Yes/No)", ylab = "Percent (%)",
        main = "Percentage of Defaulters",
        col = c("blue","red"),
        legend = rownames(tab1),
        beside = TRUE,
        ylim = c(0, 90))
text(bp, 0, round(tab1, 1),cex=1,pos=3)
```

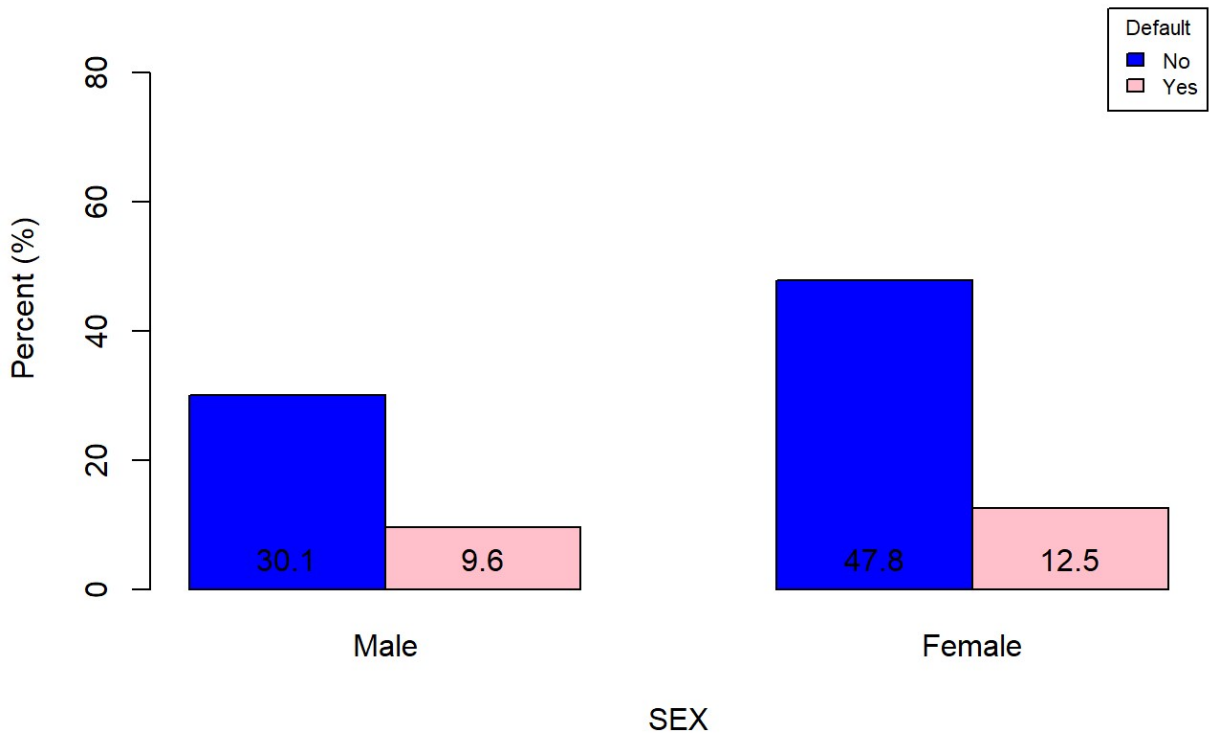


Above we observed that 22.1 % percent defaulter and 77.9% are not default cases .

```
# table for counts
gender_defaulter<- table(DEFAULT,SEX)
gender_defaulter1 <- prop.table(gender_defaulter)
gender_defaulter2 <- addmargins(gender_defaulter1)
round(gender_defaulter2*100,2)
```

```
##        SEX
## DEFAULT   Male Female    Sum
##     No   30.05  47.83  77.88
##     Yes   9.58  12.54  22.12
##     Sum  39.63  60.37 100.00
```

```
# Percentage of defaulters by Gender
tab2 <- round(prop.table(table(taiwan_bank$DEFAULT,taiwan_bank$SEX))*100,2)
# bar-plot
bp <- barplot(tab2, beside = TRUE, main = "Bar Chart For Defaulters By Gender",
col = c("blue", "pink"),
xlab = "SEX",
ylab = "Percent (%)", legend = c("No", "Yes"),
args.legend = list(title = "Default", x = "topright", cex = .7), ylim = c(0, 90))
text(bp, 0, round(tab2, 1),cex=1,pos=3)
```



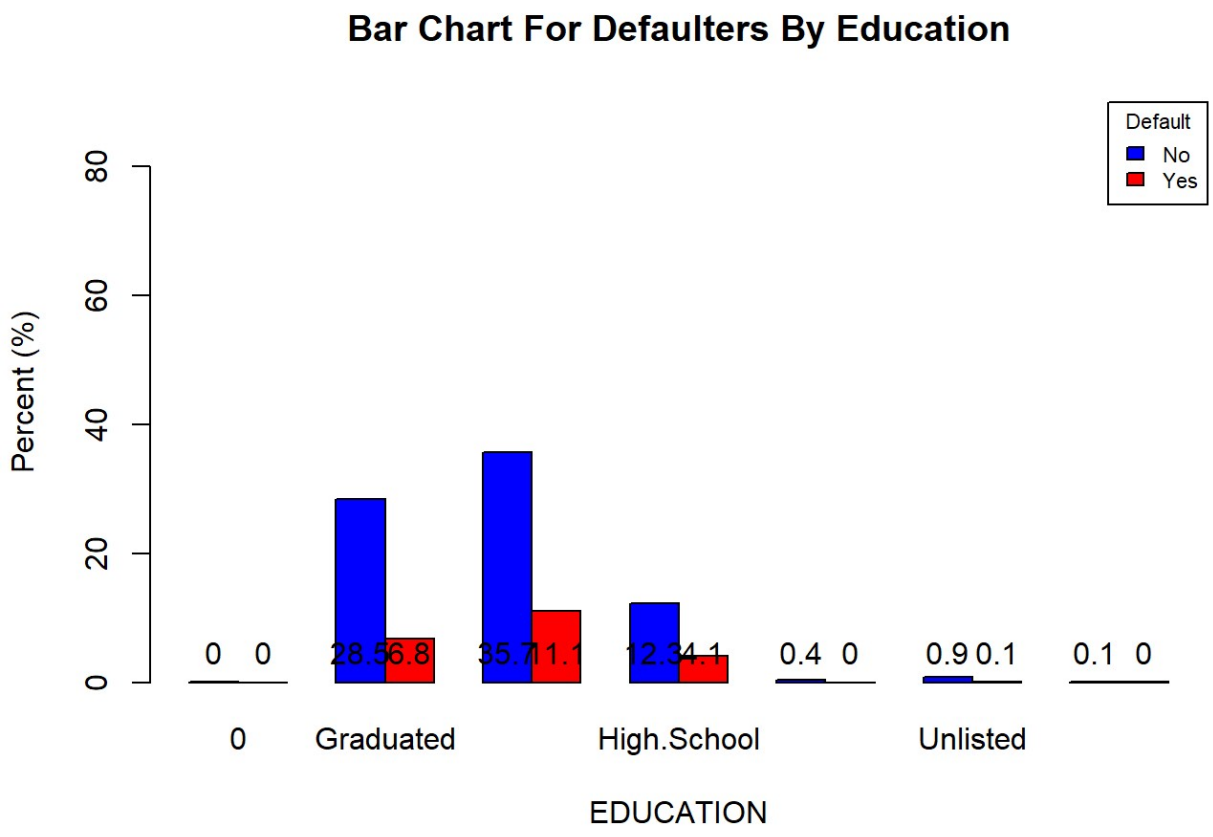
we found here more females customer are into default category whereas male cusotmer are 9.6% and female category shows 12.5%

```
# percentage of defaulters by Education
eudcation_defaulter1<- table(DEFAULT,EDUCATION)
eudcation_defaulter2 <- prop.table(eudcation_defaulter1)
eudcation_defaulter3 <- addmargins(eudcation_defaulter2)
round(eudcation_defaulter3*100,2)
```

```
##         EDUCATION
## DEFAULT      0 Graduated University High.School Others Unlisted
##     No    0.05     28.50      35.67       12.27   0.39     0.87
##     Yes   0.00      6.79      11.10        4.12   0.02     0.06
##     Sum   0.05     35.28      46.77       16.39   0.41     0.93
##         EDUCATION
## DEFAULT Unlisted_one    Sum
##     No          0.14  77.88
##     Yes         0.03  22.12
##     Sum         0.17 100.00
```

```
tab3 <- round(prop.table(table(taiwan_bank$DEFAULT,taiwan_bank$EDUCATION))*100,2)

bp <- barplot(tab3, beside = TRUE, main = "Bar Chart For Defaulters By Education",
col = c("blue", "red"),
xlab = "EDUCATION",
ylab = "Percent (%)", legend = c("No", "Yes"),
args.legend = list(title = "Default", x = "topright", cex = .7), ylim = c(0, 90))
text(bp, 0, round(tab3, 1),cex=1,pos=3)
```



Realized the customers who has been completed univeristy level - graduate or PG is more inot default side .

Checking the percentage of defaulters by Education with Gender

```
# percentage of defaulters by Education with Gender
eudcationsex_defaulter1<- table(DEFAULT,EDUCATION, SEX)
eudcationsex_defaulter2 <- prop.table(eudcationsex_defaulter1)
eudcationsex_defaulter3 <- addmargins(eudcationsex_defaulter2)
round(eudcationsex_defaulter3*100,2)
```

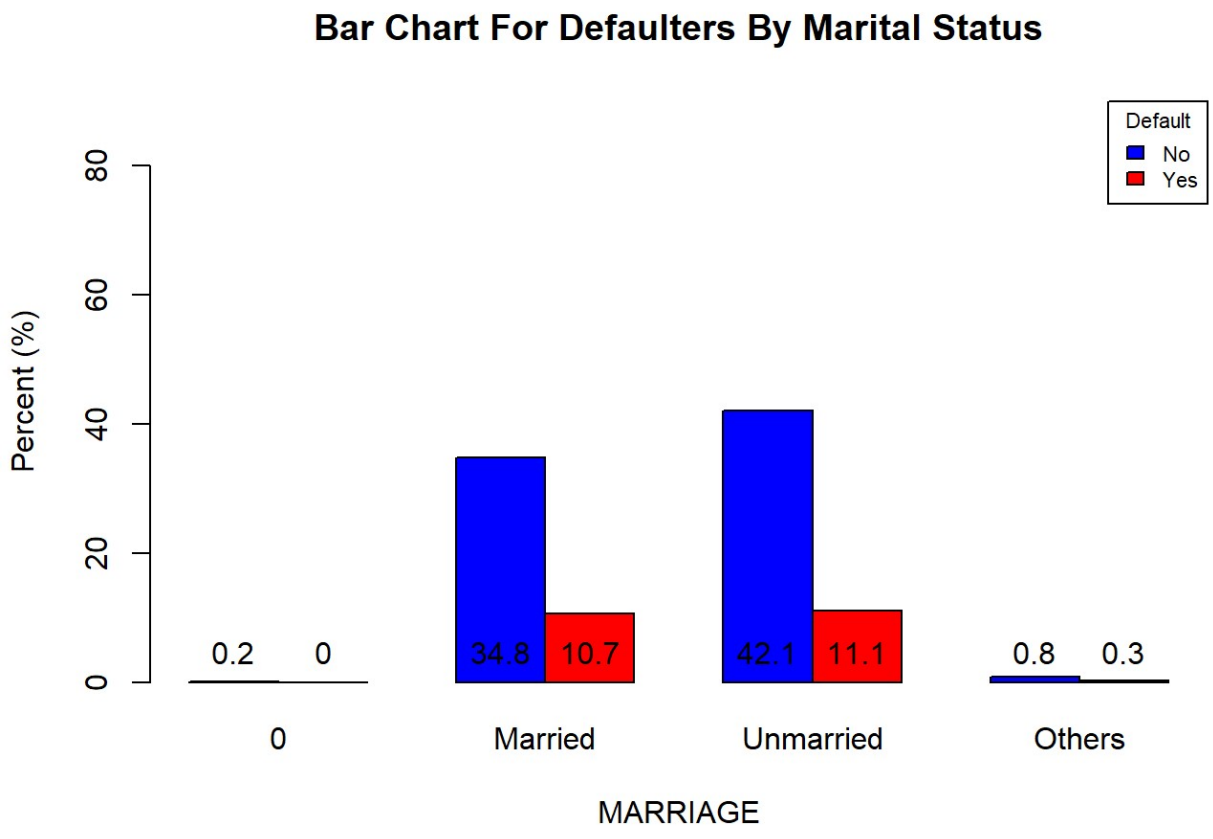
```
## , , SEX = Male
##
##        EDUCATION
## DEFAULT      0 Graduated University High.School Others Unlisted
##     No    0.03     11.49      13.22        4.82   0.13     0.30
##     Yes   0.00      3.02       4.69        1.82   0.01     0.02
##     Sum   0.03     14.51      17.91        6.63   0.14     0.32
##        EDUCATION
## DEFAULT Unlisted_one    Sum
##     No          0.07  30.05
##     Yes         0.01   9.58
##     Sum         0.08  39.63
##
## , , SEX = Female
##
##        EDUCATION
## DEFAULT      0 Graduated University High.School Others Unlisted
##     No    0.02     17.00      22.45        7.45   0.26     0.58
##     Yes   0.00      3.77       6.41        2.31   0.01     0.04
##     Sum   0.02     20.77      28.85        9.76   0.27     0.62
##        EDUCATION
## DEFAULT Unlisted_one    Sum
##     No          0.07  47.83
##     Yes         0.01  12.54
##     Sum         0.09  60.37
##
## , , SEX = Sum
##
##        EDUCATION
## DEFAULT      0 Graduated University High.School Others Unlisted
##     No    0.05     28.50      35.67       12.27   0.39     0.87
##     Yes   0.00      6.79      11.10        4.12   0.02     0.06
##     Sum   0.05     35.28      46.77       16.39   0.41     0.93
##        EDUCATION
## DEFAULT Unlisted_one    Sum
##     No          0.14  77.88
##     Yes         0.03  22.12
##     Sum         0.17 100.00
```

```
# Percentage of defaulters by MaritalStatus

marrige_defaulter1 <- table(DEFAULT,MARRIAGE)
marrige_defaulter2 <- prop.table(marrige_defaulter1)
marrige_defaulter3 <- addmargins(marrige_defaulter2)
round(marrige_defaulter3*100,2)
```

```
##         MARRIAGE
## DEFAULT      0 Married Unmarried Others    Sum
##      No    0.16   34.84     42.08   0.80  77.88
##      Yes   0.02   10.69     11.14   0.28  22.12
##      Sum   0.18   45.53     53.21   1.08 100.00
```

```
#ploting defaulters by MaritalStatus
tab4 <- round(prop.table(table(taiwan_bank$DEFAULT,taiwan_bank$MARRIAGE))*100,2)
bp <- barplot(tab4, beside = TRUE, main = "Bar Chart For Defaulters By Marital Statu
s",
col = c("blue", "red"),
xlab = "MARRIAGE",
ylab = "Percent (%)", legend = c("No", "Yes"),
args.legend = list(title = "Default", x = "topright", cex = .7), ylim = c(0, 90))
text(bp, 0, round(tab4, 1),cex=1,pos=3)
```

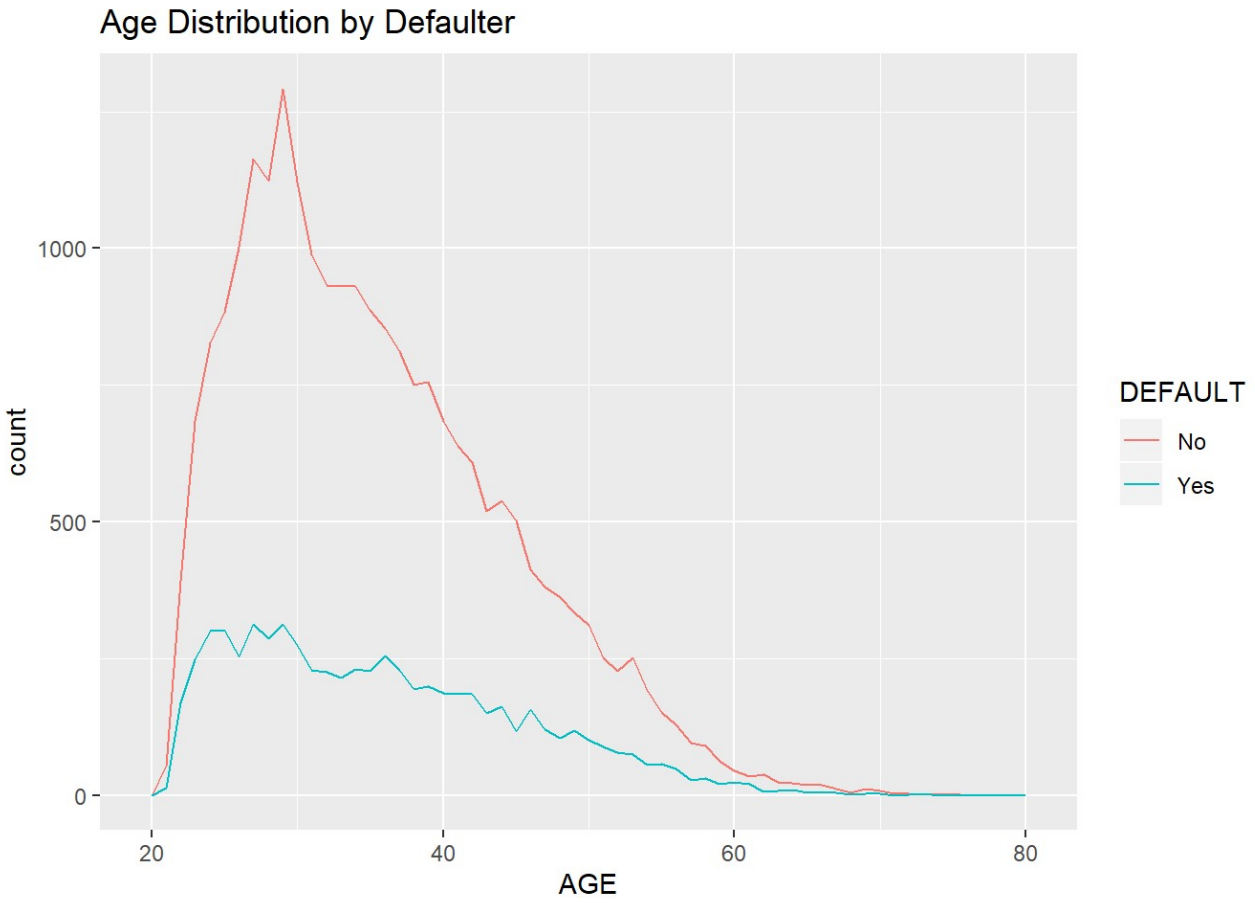


Above it shows that single are paying the bills on time ,marreid cusotmers somehow response is less comparing to unmarried customers

```
age_defaulter1 <- table(DEFAULT,AGE)
age_defaulter2 <- prop.table(age_defaulter1)
age_defaulter3 <- addmargins(age_defaulter2)
round(age_defaulter3*100,2)
```

```
##         AGE
## DEFAULT     21     22     23     24     25     26     27     28     29
##     No    0.18   1.30   2.28   2.76   2.95   3.34   3.88   3.74   4.31
##     Yes   0.05   0.56   0.82   1.00   1.01   0.84   1.04   0.95   1.04
##     Sum   0.22   1.87   3.10   3.76   3.95   4.19   4.92   4.70   5.35
##         AGE
## DEFAULT     30     31     32     33     34     35     36     37     38
##     No    3.74   3.29   3.11   3.10   3.10   2.96   2.85   2.71   2.50
##     Yes   0.91   0.76   0.75   0.72   0.77   0.75   0.85   0.76   0.65
##     Sum   4.65   4.06   3.86   3.82   3.87   3.71   3.69   3.47   3.15
##         AGE
## DEFAULT     39     40     41     42     43     44     45     46     47
##     No    2.52   2.28   2.13   2.03   1.73   1.79   1.67   1.38   1.27
##     Yes   0.66   0.62   0.62   0.62   0.50   0.54   0.39   0.52   0.40
##     Sum   3.18   2.90   2.75   2.65   2.23   2.33   2.06   1.90   1.67
##         AGE
## DEFAULT     48     49     50     51     52     53     54     55     56
##     No    1.21   1.11   1.03   0.84   0.75   0.84   0.64   0.51   0.43
##     Yes   0.35   0.40   0.34   0.29   0.26   0.25   0.19   0.19   0.16
##     Sum   1.55   1.51   1.37   1.13   1.01   1.08   0.82   0.70   0.59
##         AGE
## DEFAULT     57     58     59     60     61     62     63     64     65
##     No    0.32   0.30   0.21   0.15   0.12   0.12   0.08   0.07   0.06
##     Yes   0.09   0.10   0.07   0.08   0.07   0.02   0.03   0.03   0.02
##     Sum   0.41   0.41   0.28   0.22   0.19   0.15   0.10   0.10   0.08
##         AGE
## DEFAULT     66     67     68     69     70     71     72     73     74
##     No    0.06   0.04   0.01   0.04   0.03   0.01   0.01   0.00   0.00
##     Yes   0.02   0.02   0.00   0.01   0.01   0.00   0.00   0.01   0.00
##     Sum   0.08   0.05   0.02   0.05   0.03   0.01   0.01   0.01   0.00
##         AGE
## DEFAULT     75     79    Sum
##     No    0.01   0.00  77.88
##     Yes   0.00   0.00  22.12
##     Sum   0.01   0.00 100.00
```

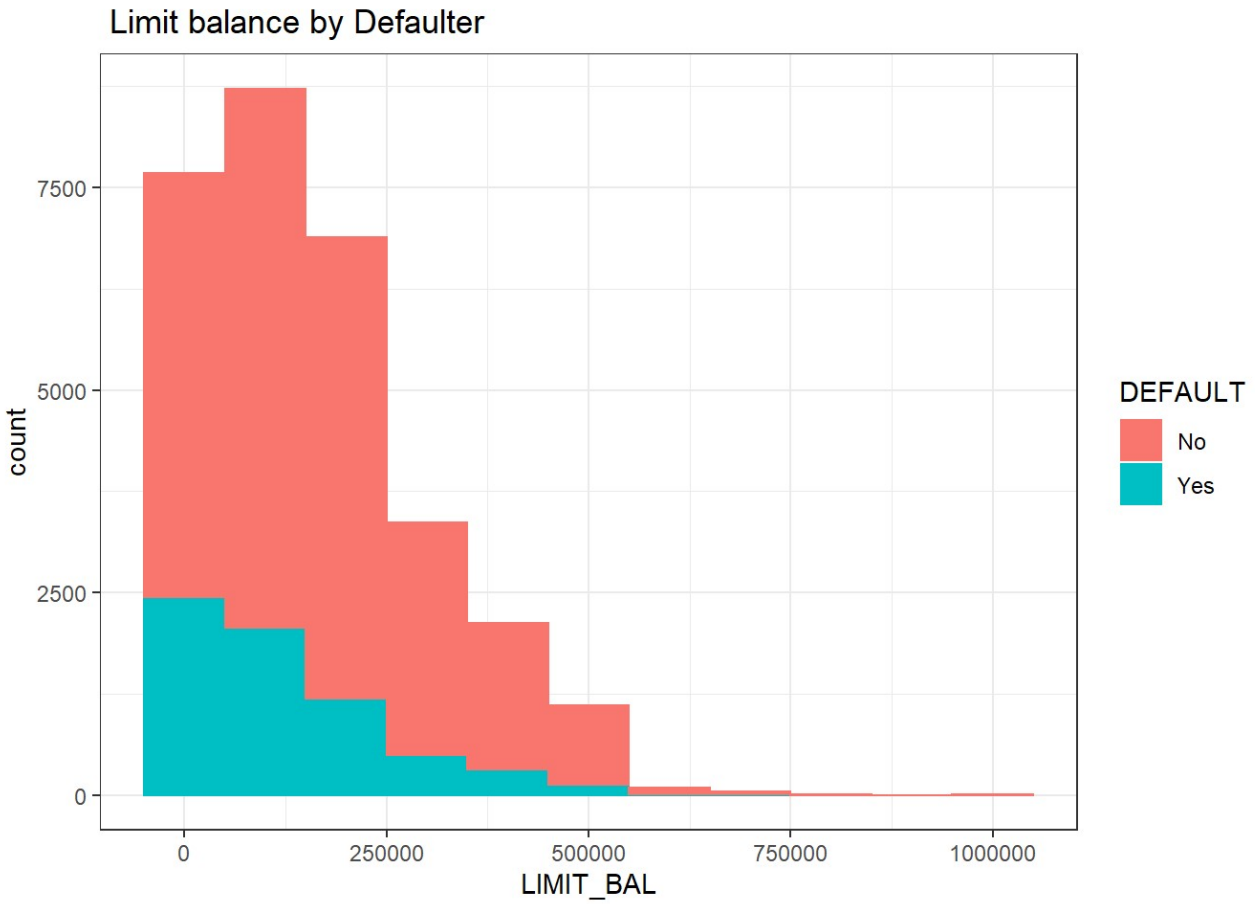
```
#plot of Age wise dafaulters
library(ggplot2)
library(dplyr)
ggplot(taiwan_bank, aes(x= AGE, colour = DEFAULT )) +
  geom_freqpoly(binwidth = 1)+ labs(title="Age Distribution by Defaulter ")
```



Observed the average age of 25 to 30 is the hightest risk , moreover customer included in theat groups, highest number of timely paid cusotmer also hailed from the same age groups only.

Plot Credit Balance with defaulters

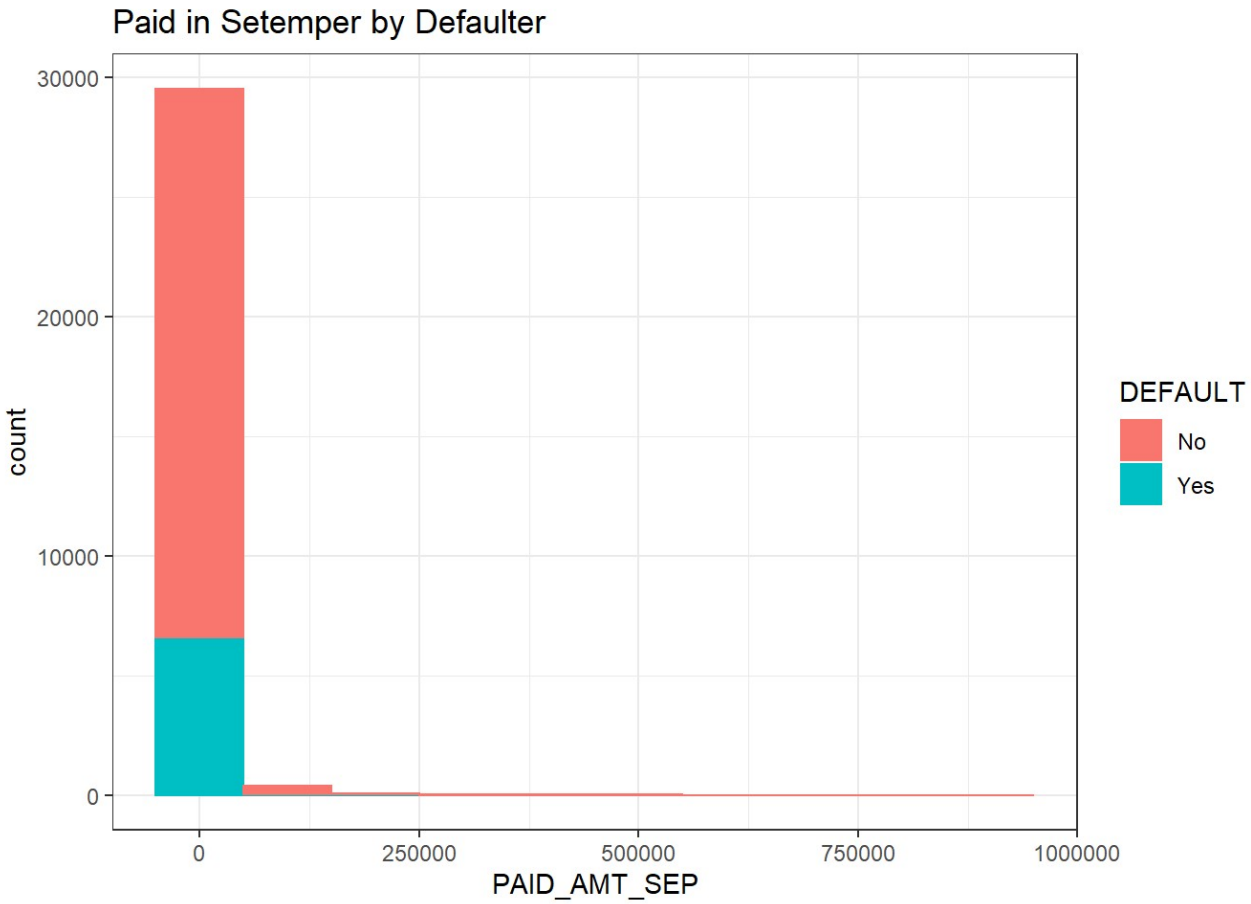
```
#Plot Credit Balance with defaulters
lb <- ggplot(taiwan_bank, aes(x=LIMIT_BAL , fill=DEFAULT, color=DEFAULT)) +
  geom_histogram(binwidth = 100000) + labs(title=" Limit balance by Defaulter")
lb + theme_bw()
```



We found customers are having credit balance limit 250000 are maximum defaulters

Amount Pid in September 2005 with defaulters

```
# Amount Pid in September 2005 with defaulters
pa <- ggplot(taiwan_bank, aes(x=PAID_AMT_SEP, fill=DEFAULT, color=DEFAULT)) +
  geom_histogram(binwidth = 100000) + labs(title="Paid in Setemper by Defaulter")
pa + theme_bw()
```



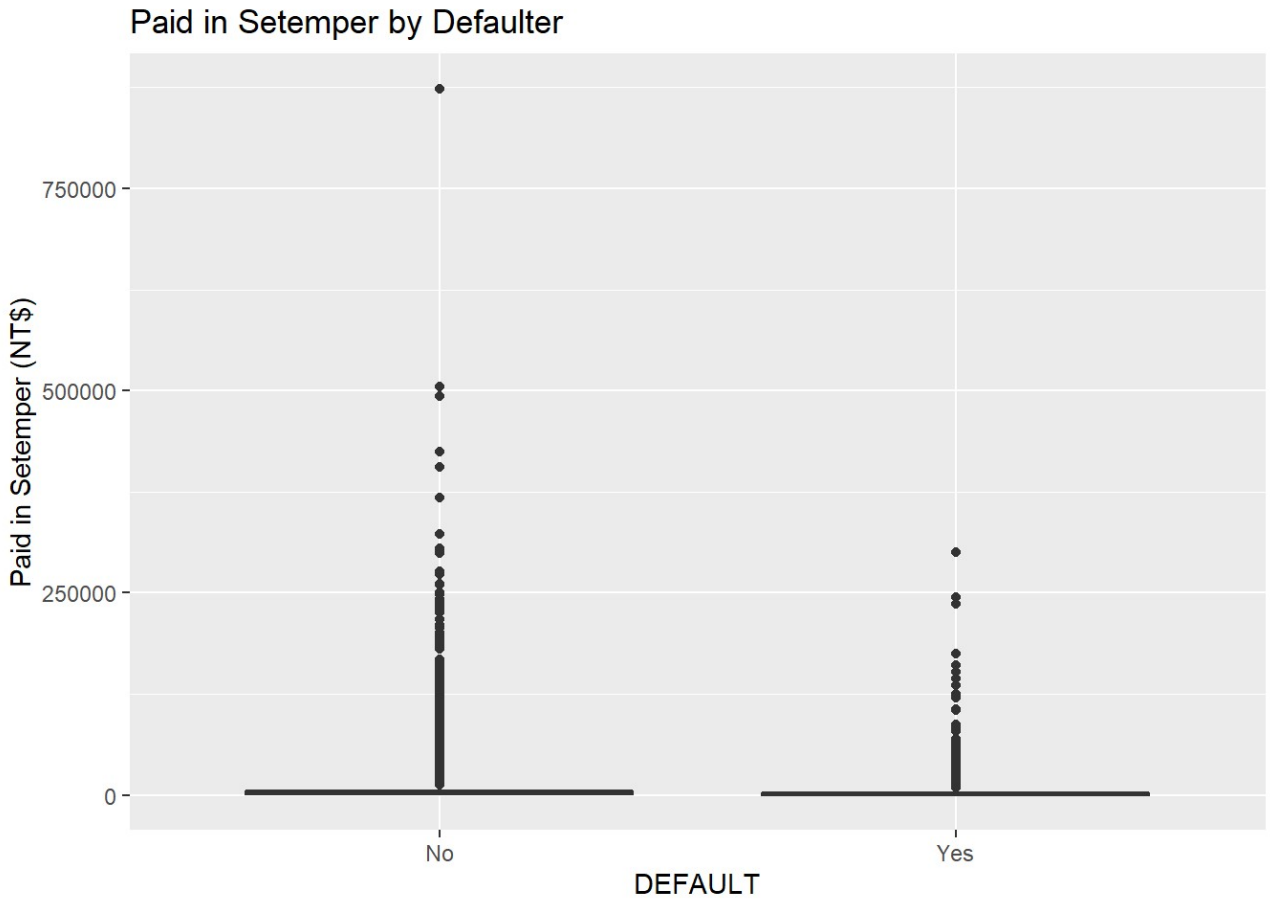
Observed herein above paid in Septemer the amount maximum customer not defaulters

plotting box plots to find the outliers

The amoount has been reapid in septemebr 2005 , we found some outlier for non defaulter customers

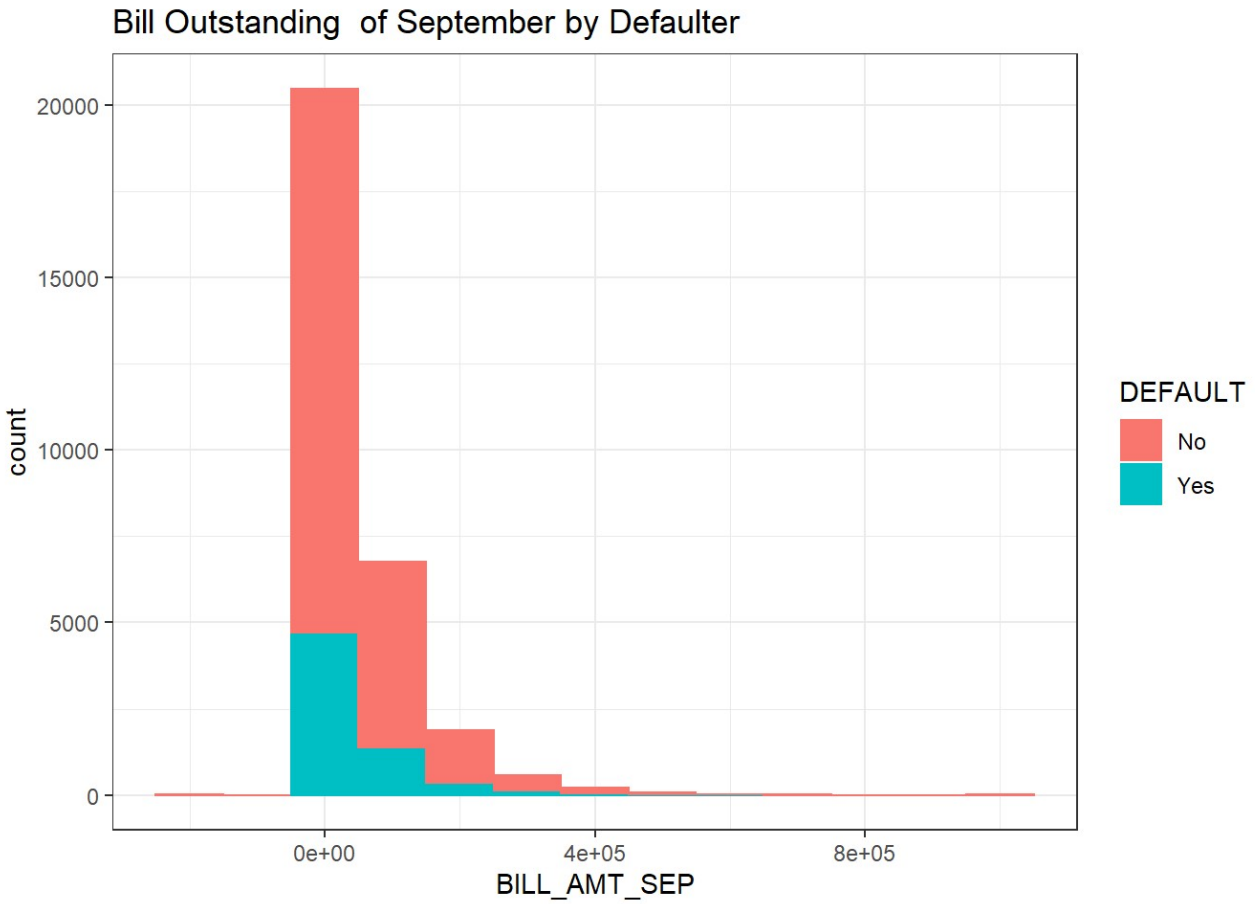
Checking with Repayment Status

```
# plotting box plots
pd <- ggplot(taiwan_bank, aes(x = DEFAULT, y = PAID_AMT_SEP)) +
  ylab("Paid in Setemper (NT$)") + geom_boxplot()
pd + labs(title="Paid in Setemper by Defaulter ")
```



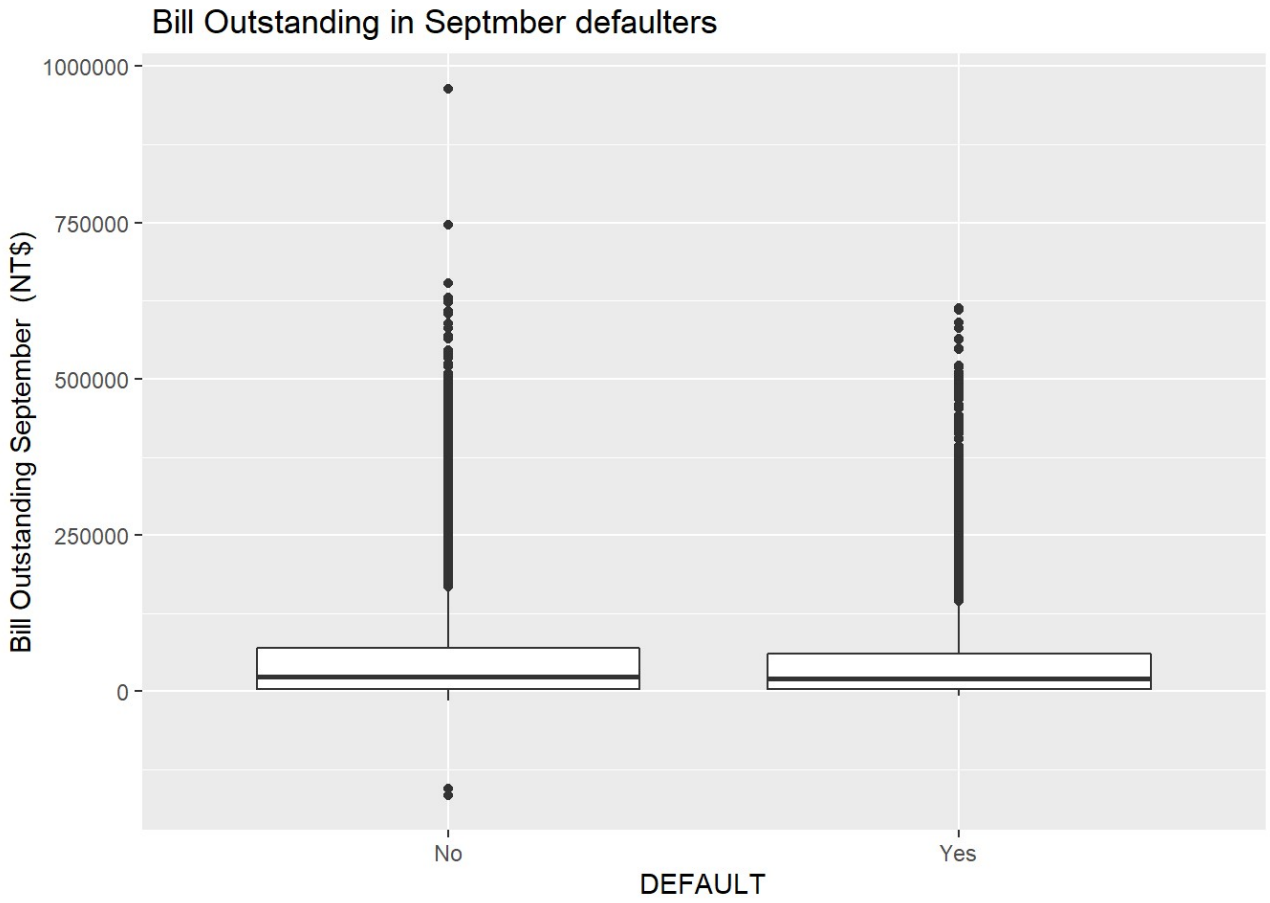
Checking bill outstanding for september with defaulters

```
bas <- ggplot(taiwan_bank, aes(x=BILL_AMT_SEP, fill= DEFAULT, color=DEFAULT)) +
  geom_histogram(binwidth = 100000) + labs(title="Bill Outstanding  of September by De
faulter ")
bas + theme_bw()
```



Checking outlier for bill outstadning amount for month of Sept 20005

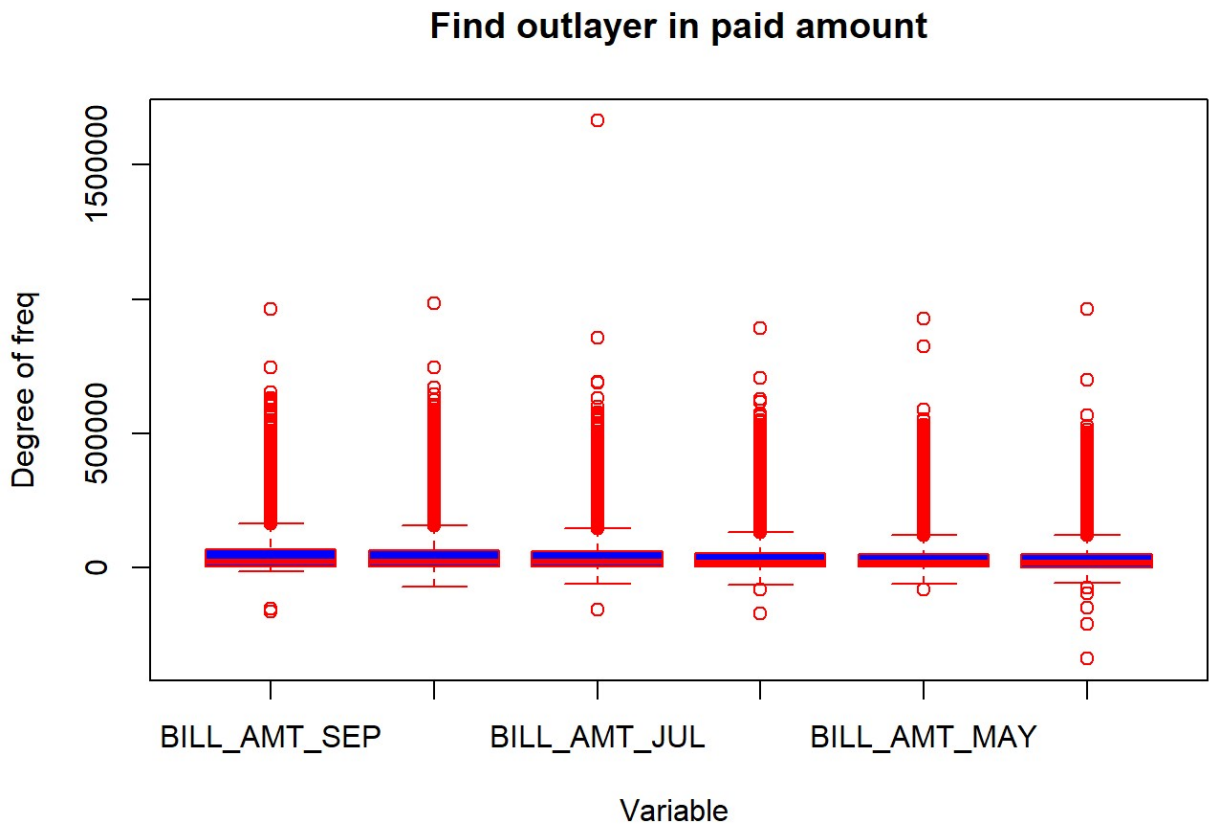
```
# plotting box plots
ba <- ggplot((taiwan_bank), aes(x = DEFAULT, y = BILL_AMT_SEP)) + ylab("Bill Outstandi
ng September  (NT$)") + geom_boxplot()
ba + labs(title=" Bill Outstanding in Septmber defaulters")
```



We got there is also some oulier bill ouststadning amounts for non -defaulter whereas for defaulters are not having outliers , believe maxiam defaulted cusotmers are under category of 50000 amount

Checking all bill outstandidng

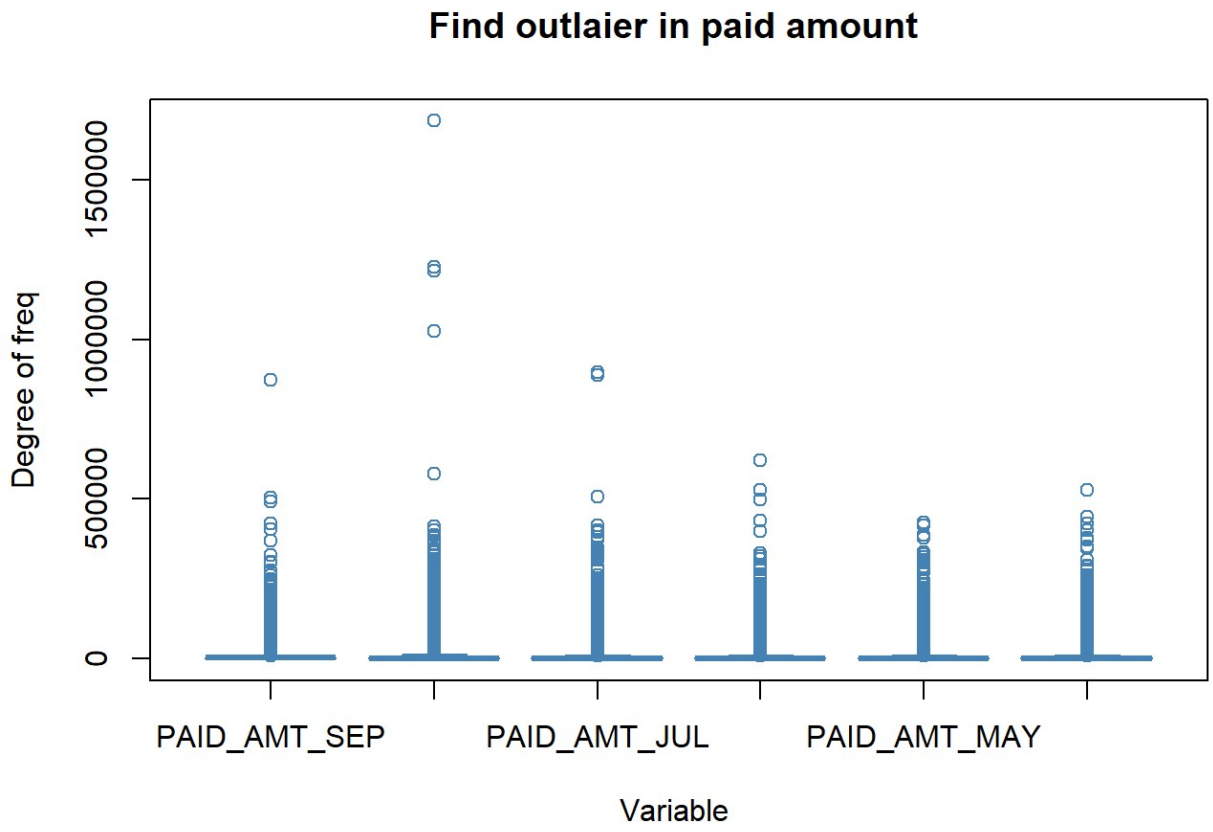
```
boxplot(taiwan_bank[,13:18],
main="Find outlayer in paid amount",
xlab="Variable ",
ylab="Degree of freq",
col="blue",
border="red" )
```



There is extereem outlier shown the out standing bill for month of July bill amount and there negative outliers found for month of APR 2005

Chekcing outliers of paid amounts

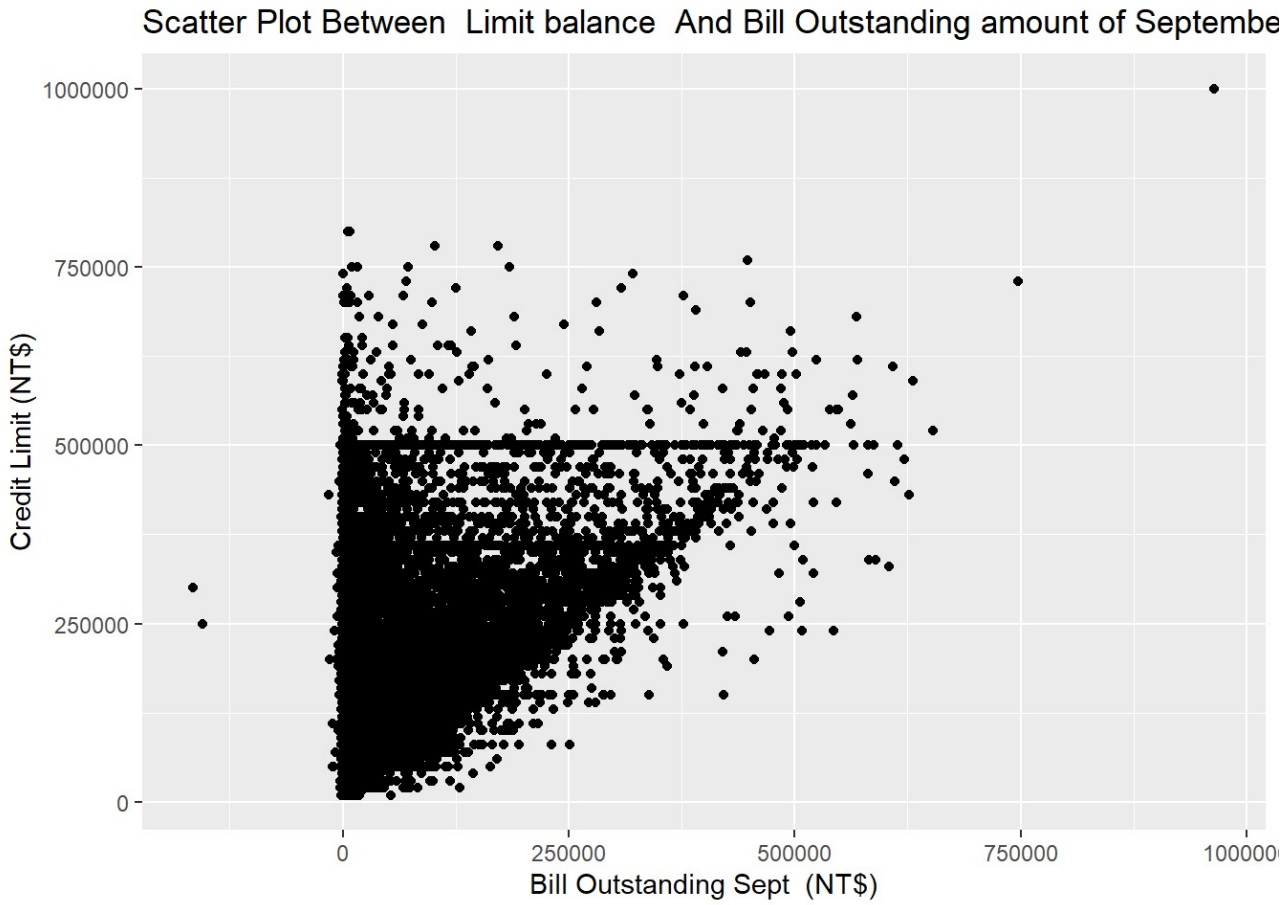
```
boxplot(taiwan_bank[,19:24],
main="Find outlaier in paid amount ",
xlab="Variable ",
ylab="Degree of freq",
col="red",
border="steelblue" )
```



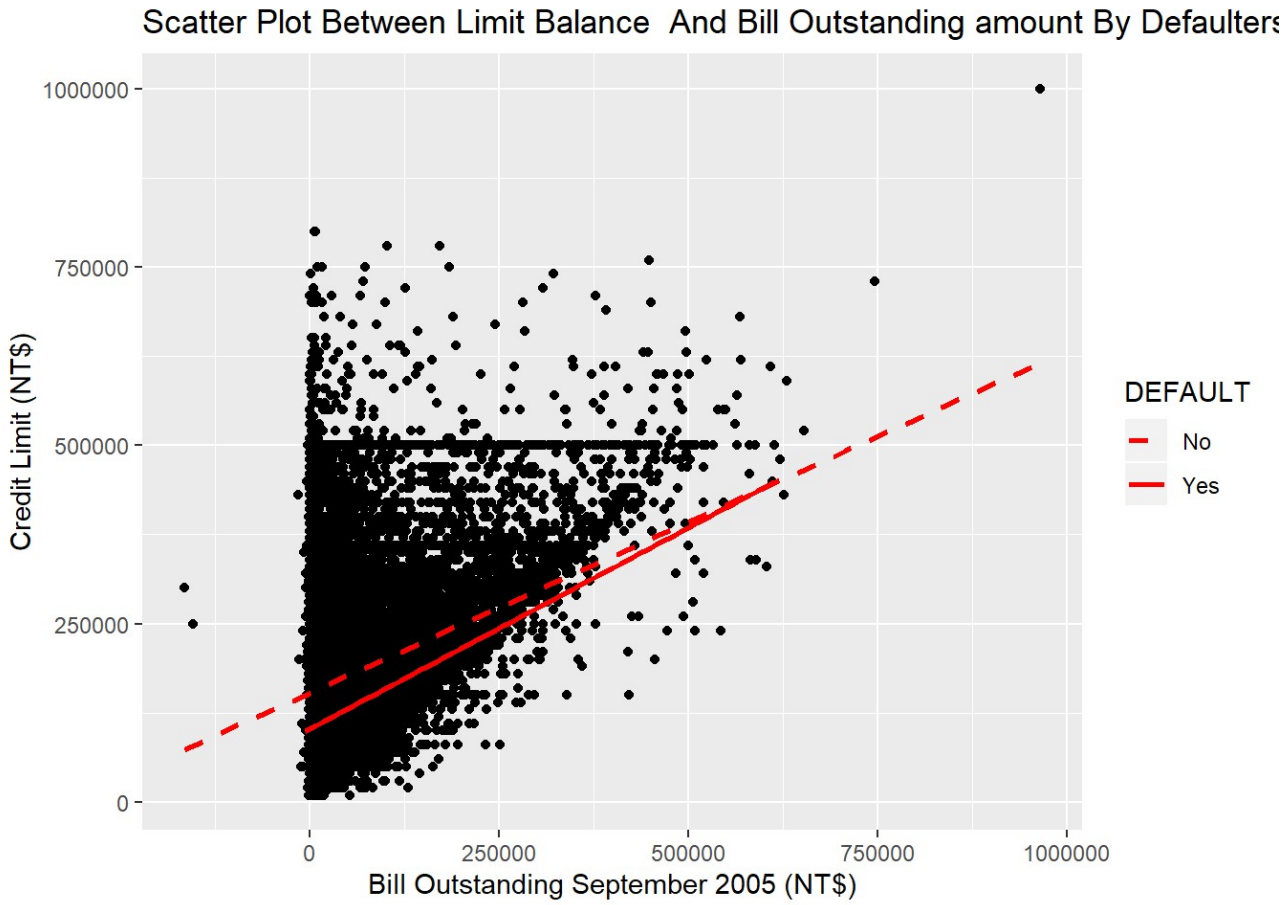
Obserged there is extreem outlier appread in the month of Aug 2005, also a minor outliers found for month Septmber and july

Checking credit balnce and out standing bill amount of Septmber 2005

```
library(ggplot2)
sept <- ggplot(taiwan_bank, aes(x = BILL_AMT_SEP, y = LIMIT_BAL)) +
  ylab("Credit Limit (NT$)") + geom_point()
sept + labs(x = "Bill Outstanding Sept  (NT$)") +
  labs(title="Scatter Plot Between  Limit balance  And Bill Outstanding amount of Sept
ember 2005")
```

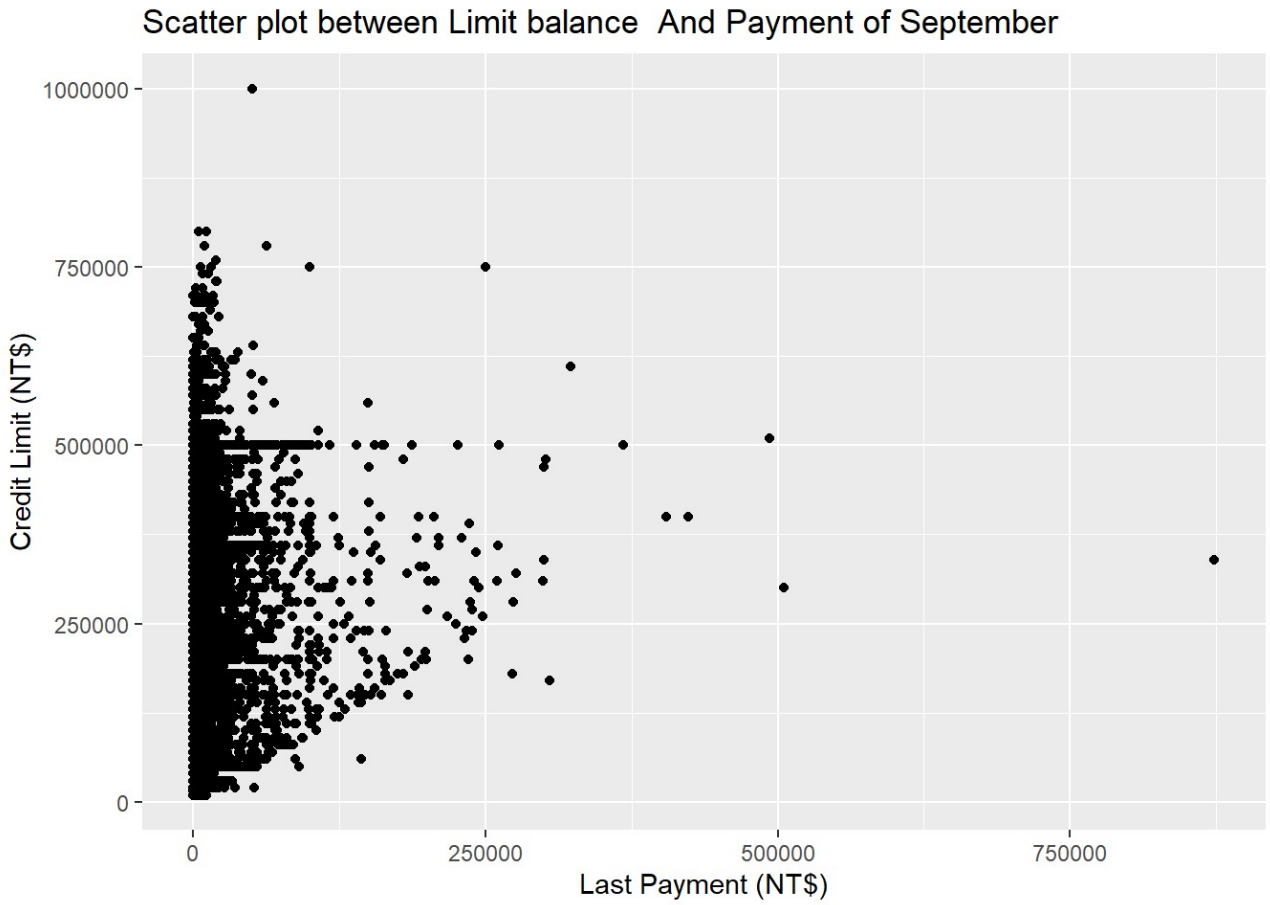



```
# plotting scatter plot
sepl <- ggplot(taiwan_bank, aes(x = BILL_AMT_SEP, y = LIMIT_BAL,linetype = DEFAULT)) +
ylab("Credit Limit (NT$)") + geom_point() + scale_linetype_manual(values=c("dashed",
"solid")) +
  geom_smooth(method=lm, se=FALSE, color= "red")
sepl + labs(x = "Bill Outstanding September 2005 (NT$)") +
labs(title="Scatter Plot Between Limit Balance  And Bill Outstanding amount By Default
ers for September 2005")
```

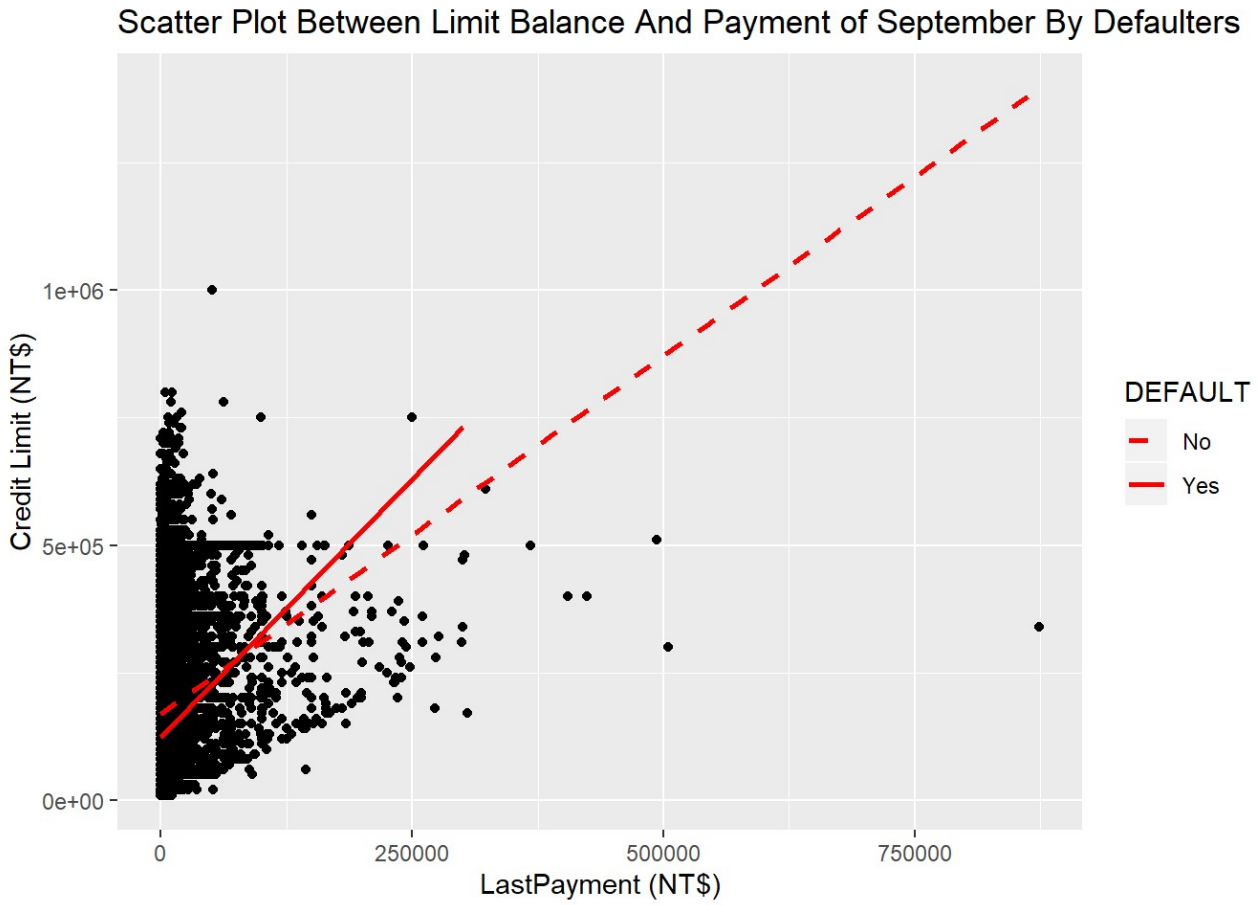


#Checking credit balnce and paid amount of Septmber 2005

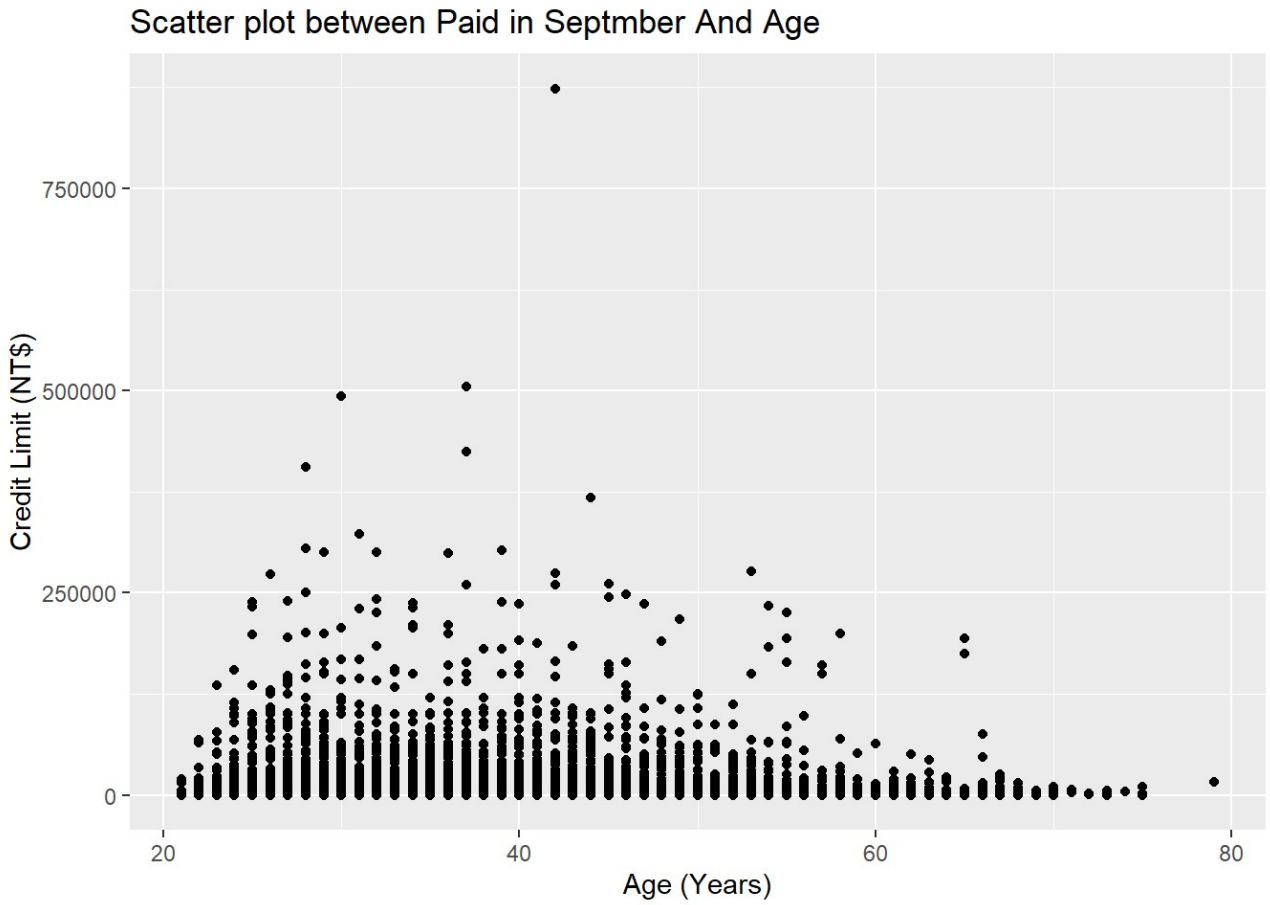
```
library(ggplot2)
# plotting scatter plot
sepp <- ggplot(taiwan_bank, aes(x = PAID_AMT_SEP, y =LIMIT_BAL)) +
  ylab("Credit Limit (NT$)") + geom_point()
sepp + labs(x = "Last Payment (NT$)") +
  labs(title="Scatter plot between Limit balance  And Payment of September")
```



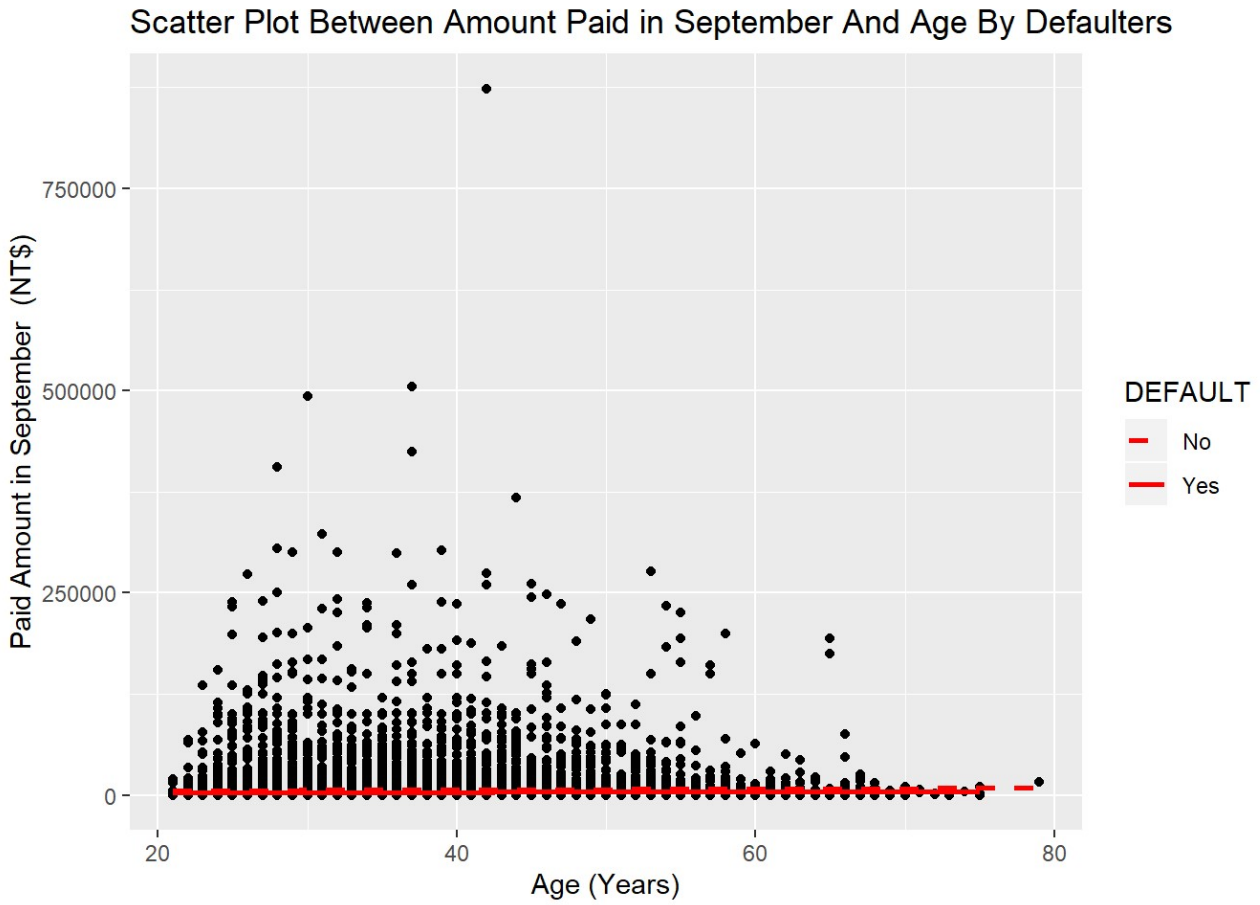

```
sepd <- ggplot(taiwan_bank, aes(x =PAID_AMT_SEP , y = LIMIT_BAL,linetype = DEFAULT))+
   ylab("Credit Limit (NT$)") + geom_point() + scale_linetype_manual(values=c("dashe
d", "solid")) +
  geom_smooth(method=lm, se=FALSE, color= "red")
sepd + labs(x = "LastPayment (NT$)") +
  labs(title="Scatter Plot Between Limit Balance And Payment of September By Defaulter
s")
```



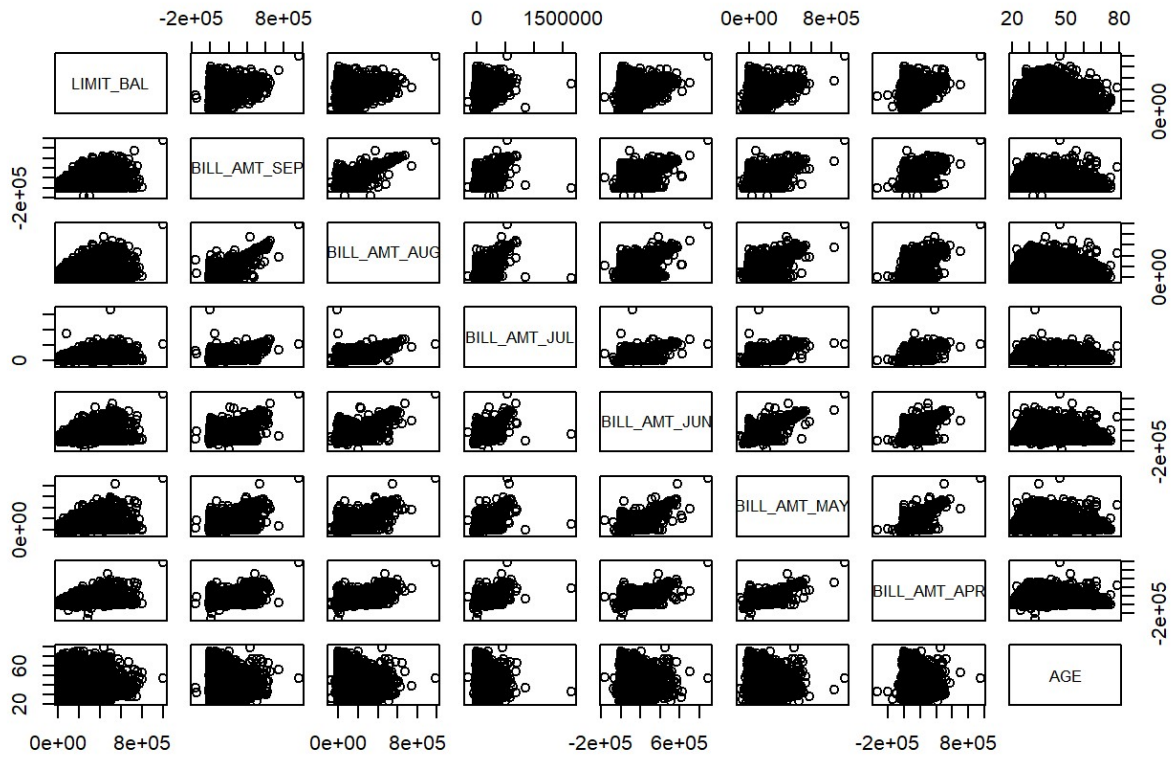

```
p <- ggplot(taiwan_bank, aes(x = AGE
, y = PAID_AMT_SEP)) +
  ylab("Credit Limit (NT$)") + geom_point()
p + labs(x = "Age (Years)") +
labs(title="Scatter plot between Paid in Septmber And Age")
```




```
p <- ggplot(taiwan_bank, aes(x = AGE, y = PAID_AMT_SEP,linetype = DEFAULT)) + ylab("Pa
id Amount in September  (NT$)") + geom_point() +
     scale_linetype_manual(values=c("dashed", "solid")) +
     geom_smooth(method=lm, se=FALSE, color= "red")
  p + labs(x = "Age (Years)") +
  labs(title="Scatter Plot Between Amount Paid in September And Age By Defaulters")
```




```
pairs(~ LIMIT_BAL + BILL_AMT_SEP + BILL_AMT_AUG +BILL_AMT_JUL+BILL_AMT_JUN +BILL_AMT_M
AY+BILL_AMT_APR + AGE ,data = taiwan_bank,
   main="Simple Scatterplot Matrix")
```




```
describe(taiwan_bank)
```

```
##               vars     n      mean        sd   median   trimmed       mad
## ID*              1 30000  15000.50   8660.40  15000.5  15000.50  11119.50
## LIMIT_BAL        2 30000 167484.32 129747.66 140000.0 151607.40 133434.00
## SEX*             3 30000      1.60      0.49      2.0      1.63      0.00
## EDUCATION*       4 30000      2.85      0.79      3.0      2.78      1.48
## MARRIAGE*        5 30000      2.55      0.52      3.0      2.55      0.00
## AGE              6 30000     35.49      9.22     34.0     34.69      8.90
## REPAY_SEP        7 30000     -0.02      1.12      0.0     -0.06      1.48
## REPAY_AUG        8 30000     -0.13      1.20      0.0     -0.20      0.00
## REPAY_JUL        9 30000     -0.17      1.20      0.0     -0.24      0.00
## REPAY_JUN       10 30000     -0.22      1.17      0.0     -0.31      0.00
## REPAY_MAY       11 30000     -0.27      1.13      0.0     -0.36      0.00
## REPAY_APR       12 30000     -0.29      1.15      0.0     -0.39      0.00
## BILL_AMT_SEP    13 30000  51223.33  73635.86  22381.5  35359.66  32321.42
## BILL_AMT_AUG    14 30000  49179.08  71173.77  21200.0  33836.10  30852.91
## BILL_AMT_JUL    15 30000  47013.15  69349.39  20088.5  32064.43  29219.82
## BILL_AMT_JUN    16 30000  43262.95  64332.86  19052.0  29212.37  27659.39
## BILL_AMT_MAY    17 30000  40311.40  60797.16  18104.5  26920.95  26224.97
## BILL_AMT_APR    18 30000  38871.76  59554.11  17071.0  25726.08  24840.96
## PAID_AMT_SEP    19 30000   5663.58  16563.28   2100.0   2997.21   2864.38
## PAID_AMT_AUG    20 30000   5921.16  23040.87   2009.0   2876.43   2951.86
## PAID_AMT_JUL    21 30000   5225.68  17606.96   1800.0   2468.91   2661.27
## PAID_AMT_JUN    22 30000   4826.08  15666.16   1500.0   2199.23   2223.90
## PAID_AMT_MAY    23 30000   4799.39  15278.31   1500.0   2202.19   2223.90
## PAID_AMT_APR    24 30000   5215.50  17777.47   1500.0   2165.33   2223.90
## DEFAULT*        25 30000      1.22      0.42      1.0      1.15      0.00
##                   min     max   range  skew kurtosis     se
## ID*                 1   30000   29999  0.00    -1.20  50.00
## LIMIT_BAL       10000 1000000  990000  0.99     0.54 749.10
## SEX*                1       2       1 -0.42    -1.82   0.00
## EDUCATION*          1       7       6  0.97     2.08   0.00
## MARRIAGE*           1       4       3 -0.02    -1.36   0.00
## AGE                21      79      58  0.73     0.04   0.05
## REPAY_SEP          -2       8      10  0.73     2.72   0.01
## REPAY_AUG          -2       8      10  0.79     1.57   0.01
## REPAY_JUL          -2       8      10  0.84     2.08   0.01
## REPAY_JUN          -2       8      10  1.00     3.50   0.01
## REPAY_MAY          -2       8      10  1.01     3.99   0.01
## REPAY_APR          -2       8      10  0.95     3.43   0.01
## BILL_AMT_SEP -165580  964511 1130091  2.66     9.80 425.14
## BILL_AMT_AUG  -69777  983931 1053708  2.70    10.30 410.92
## BILL_AMT_JUL -157264 1664089 1821353  3.09    19.78 400.39
## BILL_AMT_JUN -170000  891586 1061586  2.82    11.31 371.43
## BILL_AMT_MAY  -81334  927171 1008505  2.88    12.30 351.01
## BILL_AMT_APR -339603  961664 1301267  2.85    12.27 343.84
## PAID_AMT_SEP       0  873552  873552 14.67   415.16  95.63
## PAID_AMT_AUG       0 1684259 1684259 30.45  1641.25 133.03
## PAID_AMT_JUL       0  896040  896040 17.21   564.18 101.65
```

```
## PAID_AMT_JUN       0  621000  621000 12.90   277.27  90.45
## PAID_AMT_MAY       0  426529  426529 11.13   180.02  88.21
## PAID_AMT_APR       0  528666  528666 10.64   167.12 102.64
## DEFAULT*           1       2       1  1.34    -0.20   0.00
```

```
taiwan_bank2 <- subset(taiwan_bank, select = c(13:24))
repay_status <- (subset(taiwan_bank, select = c(7:12)))
```

Check Multicollinearity for numerical variable

```
#Correlation /Check Multicollinearity for numerical variables
taiwan_bank4 <- subset(taiwan_data, select = c(2,6:25))
str(taiwan_bank4)
```

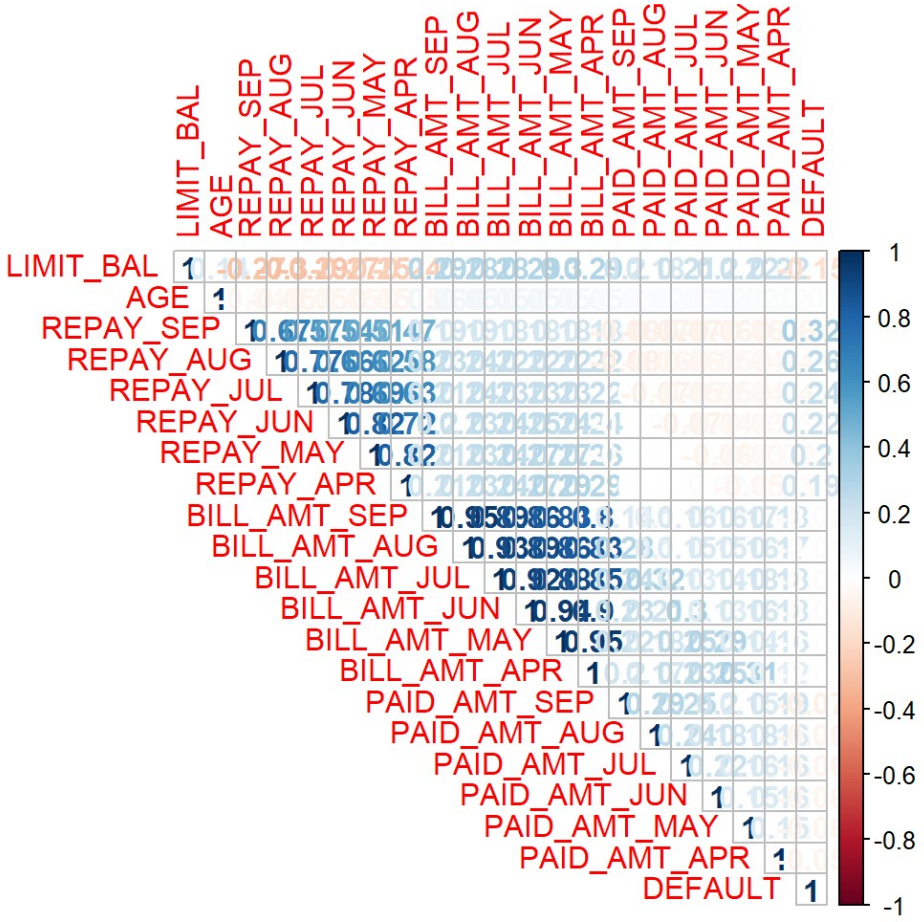
```
## Classes 'tbl_df', 'tbl' and 'data.frame':    30000 obs. of  21 variables:
##  $ LIMIT_BAL   : num  20000 120000 90000 50000 50000 50000 500000 100000 140000 200
00 ...
##  $ AGE         : num  24 26 34 37 57 37 29 23 28 35 ...
##  $ REPAY_SEP   : num  2 -1 0 0 -1 0 0 0 0 -2 ...
##  $ REPAY_AUG   : num  2 2 0 0 0 0 0 -1 0 -2 ...
##  $ REPAY_JUL   : num  -1 0 0 0 -1 0 0 -1 2 -2 ...
##  $ REPAY_JUN   : num  -1 0 0 0 0 0 0 0 0 -2 ...
##  $ REPAY_MAY   : num  -2 0 0 0 0 0 0 0 0 -1 ...
##  $ REPAY_APR   : num  -2 2 0 0 0 0 0 -1 0 -1 ...
##  $ BILL_AMT_SEP: num  3913 2682 29239 46990 8617 ...
##  $ BILL_AMT_AUG: num  3102 1725 14027 48233 5670 ...
##  $ BILL_AMT_JUL: num  689 2682 13559 49291 35835 ...
##  $ BILL_AMT_JUN: num  0 3272 14331 28314 20940 ...
##  $ BILL_AMT_MAY: num  0 3455 14948 28959 19146 ...
##  $ BILL_AMT_APR: num  0 3261 15549 29547 19131 ...
##  $ PAID_AMT_SEP: num  0 0 1518 2000 2000 ...
##  $ PAID_AMT_AUG: num  689 1000 1500 2019 36681 ...
##  $ PAID_AMT_JUL: num  0 1000 1000 1200 10000 657 38000 0 432 0 ...
##  $ PAID_AMT_JUN: num  0 1000 1000 1100 9000 ...
##  $ PAID_AMT_MAY: num  0 0 1000 1069 689 ...
##  $ PAID_AMT_APR: num  0 2000 5000 1000 679 ...
##  $ DEFAULT     : num  1 1 0 0 0 0 0 0 0 0 ...
```

```
#ggpairs(taiwan_bank4)
```

```
#Checking cor plot#
taiwan_bank4 = as.data.frame(taiwan_bank4)
taiwan_bank_matrix = cor(taiwan_bank4)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
corrplot(taiwan_bank_matrix, type="upper", method="number")
```




```
ml <- lm(DEFAULT ~., data=taiwan_bank4)
summary(ml)
```

```
##
## Call:
## lm(formula = DEFAULT ~ ., data = taiwan_bank4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29702 -0.23979 -0.16282  0.03012  1.28701
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.094e-01  9.142e-03  22.905  < 2e-16 ***
## LIMIT_BAL    -6.180e-08  2.102e-08  -2.940  0.00328 **
## AGE           1.761e-03  2.464e-04   7.145 9.18e-13 ***
## REPAY_SEP     9.580e-02  2.769e-03  34.604  < 2e-16 ***
## REPAY_AUG     1.969e-02  3.340e-03   5.896 3.77e-09 ***
## REPAY_JUL     1.166e-02  3.588e-03   3.249  0.00116 **
## REPAY_JUN     3.199e-03  3.977e-03   0.804  0.42114
## REPAY_MAY     5.603e-03  4.307e-03   1.301  0.19334
## REPAY_APR     8.817e-04  3.523e-03   0.250  0.80238
## BILL_AMT_SEP -6.377e-07  1.142e-07  -5.586 2.35e-08 ***
## BILL_AMT_AUG  1.581e-07  1.605e-07   0.986  0.32437
## BILL_AMT_JUL  2.389e-08  1.511e-07   0.158  0.87436
## BILL_AMT_JUN -6.386e-08  1.575e-07  -0.406  0.68507
## BILL_AMT_MAY -1.143e-08  1.847e-07  -0.062  0.95064
## BILL_AMT_APR  1.132e-07  1.462e-07   0.774  0.43881
## PAID_AMT_SEP -7.466e-07  1.772e-07  -4.214 2.52e-05 ***
## PAID_AMT_AUG -2.103e-07  1.458e-07  -1.443  0.14906
## PAID_AMT_JUL -3.881e-08  1.690e-07  -0.230  0.81840
## PAID_AMT_JUN -2.538e-07  1.840e-07  -1.379  0.16785
## PAID_AMT_MAY -3.485e-07  1.909e-07  -1.825  0.06802 .
## PAID_AMT_APR -1.030e-07  1.366e-07  -0.754  0.45071
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.389 on 29979 degrees of freedom
## Multiple R-squared:  0.1224, Adjusted R-squared:  0.1218
## F-statistic: 209.1 on 20 and 29979 DF,  p-value: < 2.2e-16
```

Found there is high level of linear correlations between the amount of bill statements in different months.we have to VIF techinic to chekc the multicollinearity before preparing the final models for analyzation but as droping some variable is not the best decision right now.

Checking correlation of categorical data

We will perform Chi-Square Test to check he null hypothesis whether this pay status variable is independent or not, if p-value less than 0.05 significance than we reject the null hypothesis that variables are independent.

```
describe(repay_status)
```

```
##           vars     n  mean   sd median trimmed  mad min max range skew
## REPAY_SEP    1 30000 -0.02 1.12      0   -0.06 1.48  -2   8    10 0.73
## REPAY_AUG    2 30000 -0.13 1.20      0   -0.20 0.00  -2   8    10 0.79
## REPAY_JUL    3 30000 -0.17 1.20      0   -0.24 0.00  -2   8    10 0.84
## REPAY_JUN    4 30000 -0.22 1.17      0   -0.31 0.00  -2   8    10 1.00
## REPAY_MAY    5 30000 -0.27 1.13      0   -0.36 0.00  -2   8    10 1.01
## REPAY_APR    6 30000 -0.29 1.15      0   -0.39 0.00  -2   8    10 0.95
##           kurtosis   se
## REPAY_SEP     2.72 0.01
## REPAY_AUG     1.57 0.01
## REPAY_JUL     2.08 0.01
## REPAY_JUN     3.50 0.01
## REPAY_MAY     3.99 0.01
## REPAY_APR     3.43 0.01
```

```
sep <- chisq.test(REPAY_APR,REPAY_SEP)
```

```
## Warning in chisq.test(REPAY_APR, REPAY_SEP): Chi-squared approximation may
## be incorrect
```

```
aug <- chisq.test(REPAY_APR,REPAY_AUG)
```

```
## Warning in chisq.test(REPAY_APR, REPAY_AUG): Chi-squared approximation may
## be incorrect
```

```
jul <- chisq.test(REPAY_APR,REPAY_JUL)
```

```
## Warning in chisq.test(REPAY_APR, REPAY_JUL): Chi-squared approximation may
## be incorrect
```

```
jun <- chisq.test(REPAY_APR,REPAY_JUN)
```

```
## Warning in chisq.test(REPAY_APR, REPAY_JUN): Chi-squared approximation may
## be incorrect
```

```
may <- chisq.test(REPAY_APR,REPAY_MAY)
```

```
## Warning in chisq.test(REPAY_APR, REPAY_MAY): Chi-squared approximation may
## be incorrect
```

```
apr <- chisq.test(REPAY_APR,REPAY_APR)
```

```
## Warning in chisq.test(REPAY_APR, REPAY_APR): Chi-squared approximation may
## be incorrect
```

```
sep
```

```
##
##  Pearson's Chi-squared test
##
## data:  REPAY_APR and REPAY_SEP
## X-squared = 26637, df = 90, p-value < 2.2e-16
```

```
aug
```

```
##
##  Pearson's Chi-squared test
##
## data:  REPAY_APR and REPAY_AUG
## X-squared = 29864, df = 90, p-value < 2.2e-16
```

```
jul
```

```
##
##  Pearson's Chi-squared test
##
## data:  REPAY_APR and REPAY_JUL
## X-squared = 51610, df = 90, p-value < 2.2e-16
```

```
jun
```

```
##
##  Pearson's Chi-squared test
##
## data:  REPAY_APR and REPAY_JUN
## X-squared = 81782, df = 90, p-value < 2.2e-16
```

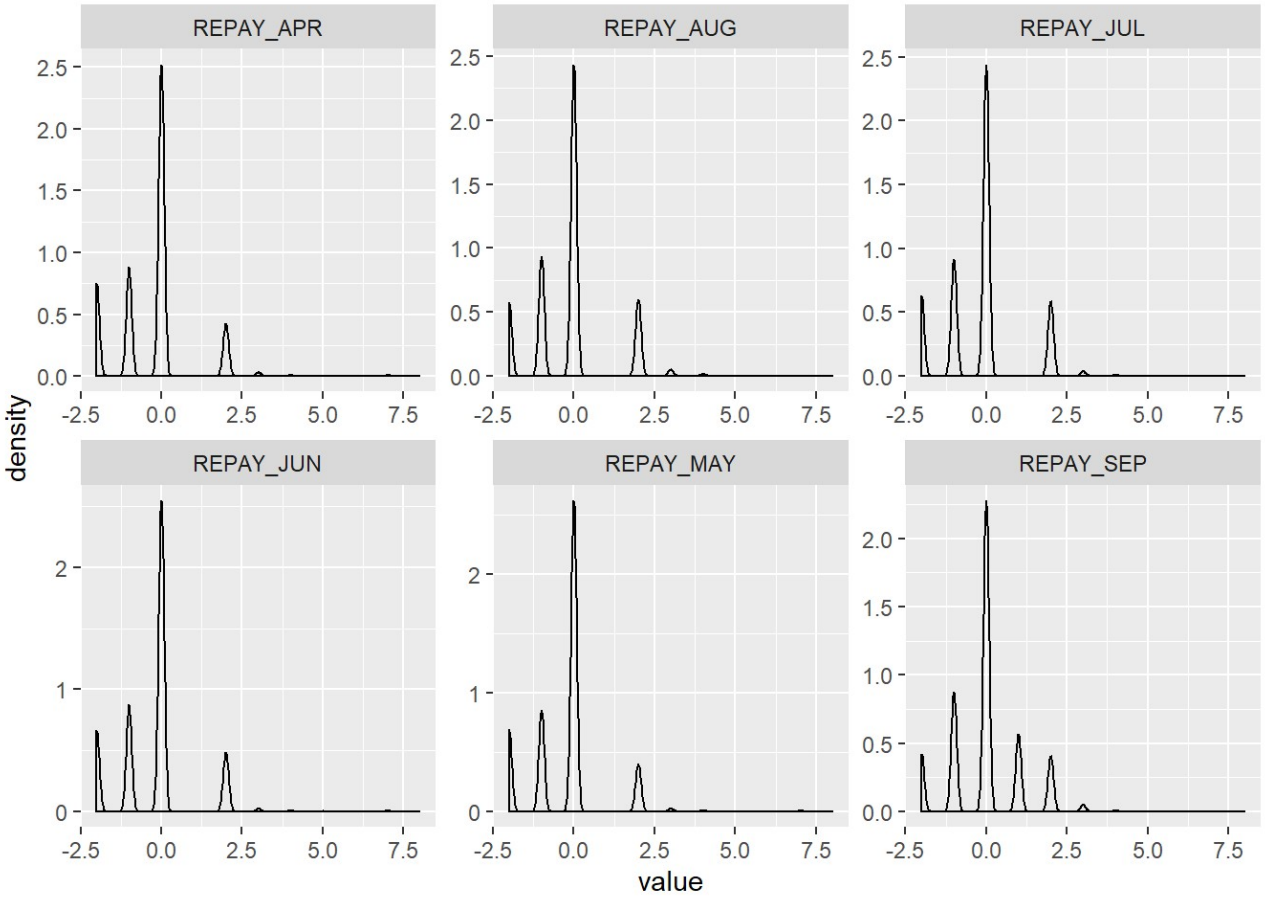
```
may
```

```
##
##  Pearson's Chi-squared test
##
## data:  REPAY_APR and REPAY_MAY
## X-squared = 114071, df = 81, p-value < 2.2e-16
```

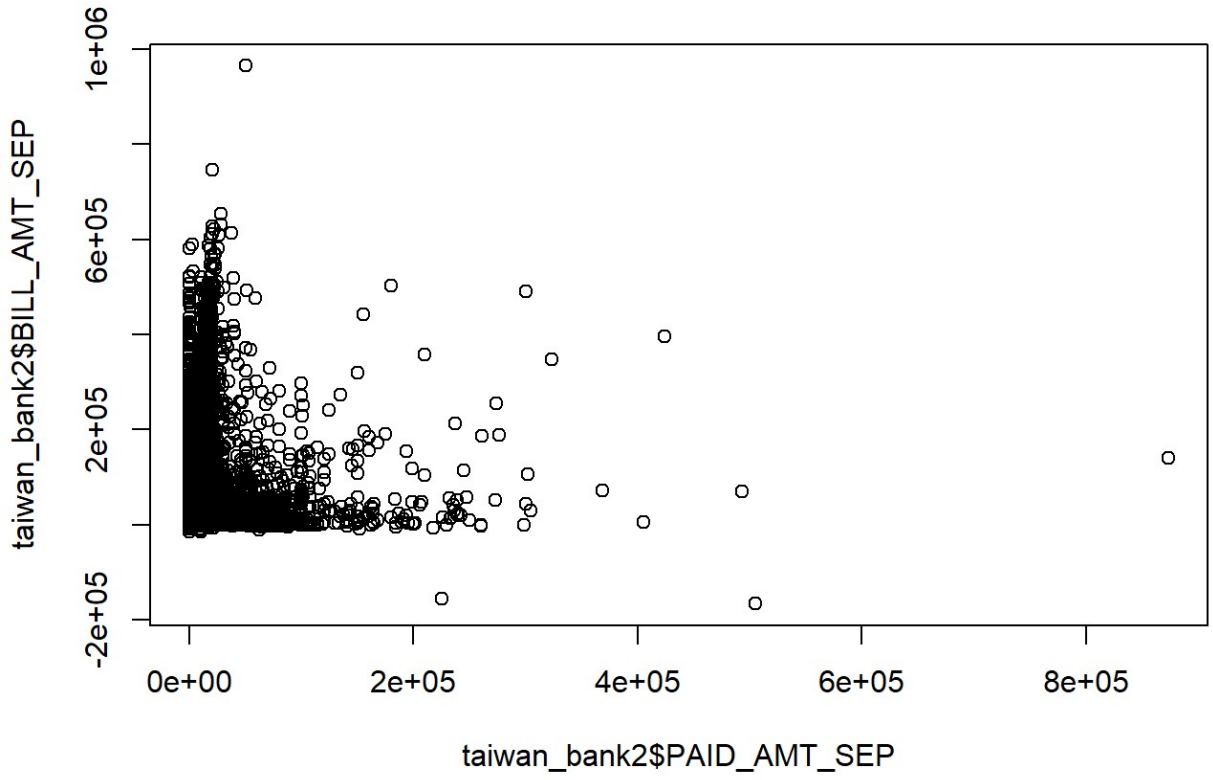
We found the pay status vairables are hihglty correlated to each other , checked only for April repaysment stauts randomely . It looks like these pay status categorical variables are dependent on each other and impact of REPAY_ SEP to REPAY_APR variables to default.payment DEFAULT is high.

Varibale REPAY_APR is a categorical variables with the levels: -1=pay duly, 1=payment delay for one month, 2=payment delay for two months, . 8=payment delay for eight months, 9=payment delay for nine months and above. Let's look at real REPAY _APR columns

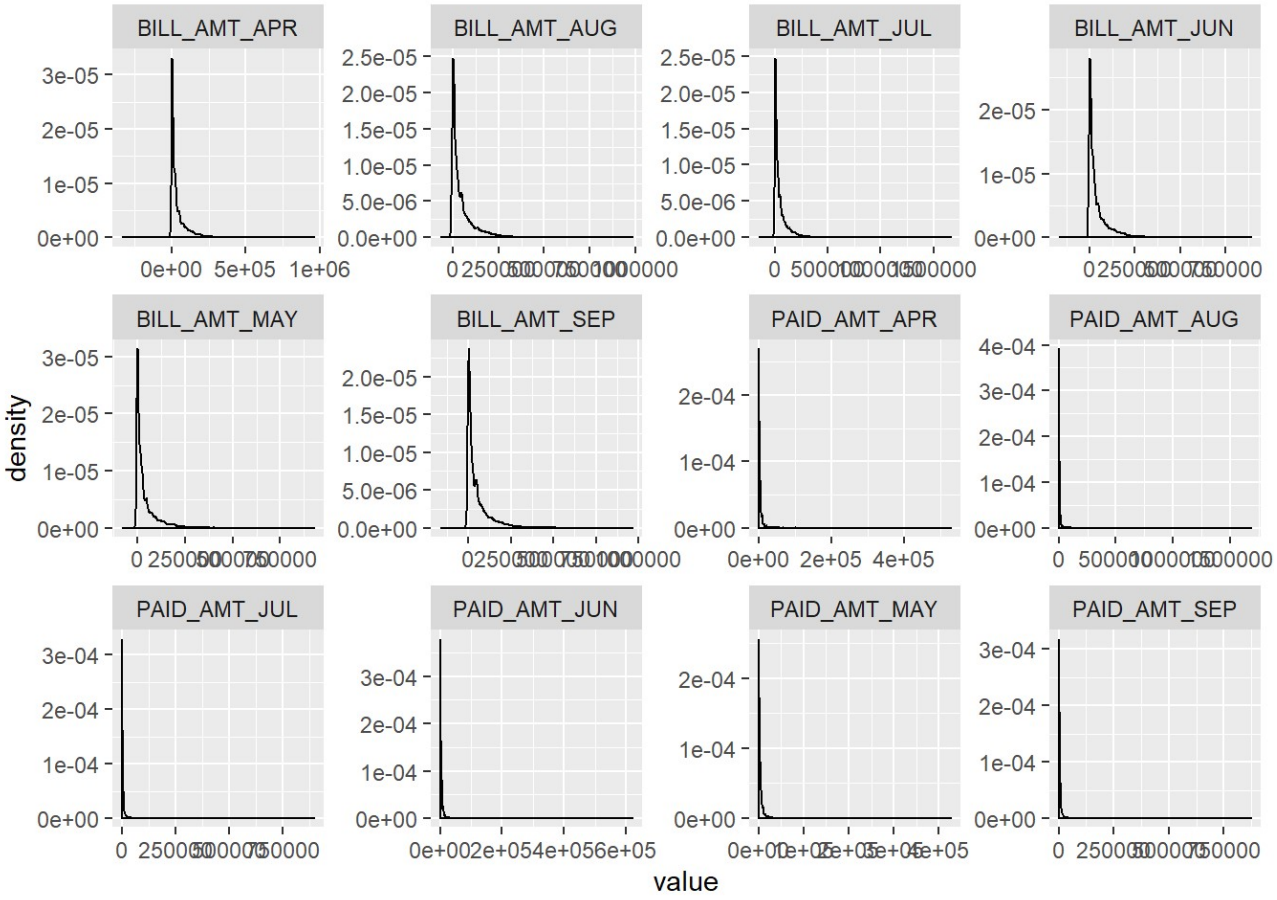
```
library(ggplot2)
  repay_status%>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_density()
```




```
plot(taiwan_bank2$PAID_AMT_SEP,taiwan_bank2$BILL_AMT_SEP)
```




```
library(ggplot2)
taiwan_bank2 %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_density()
```

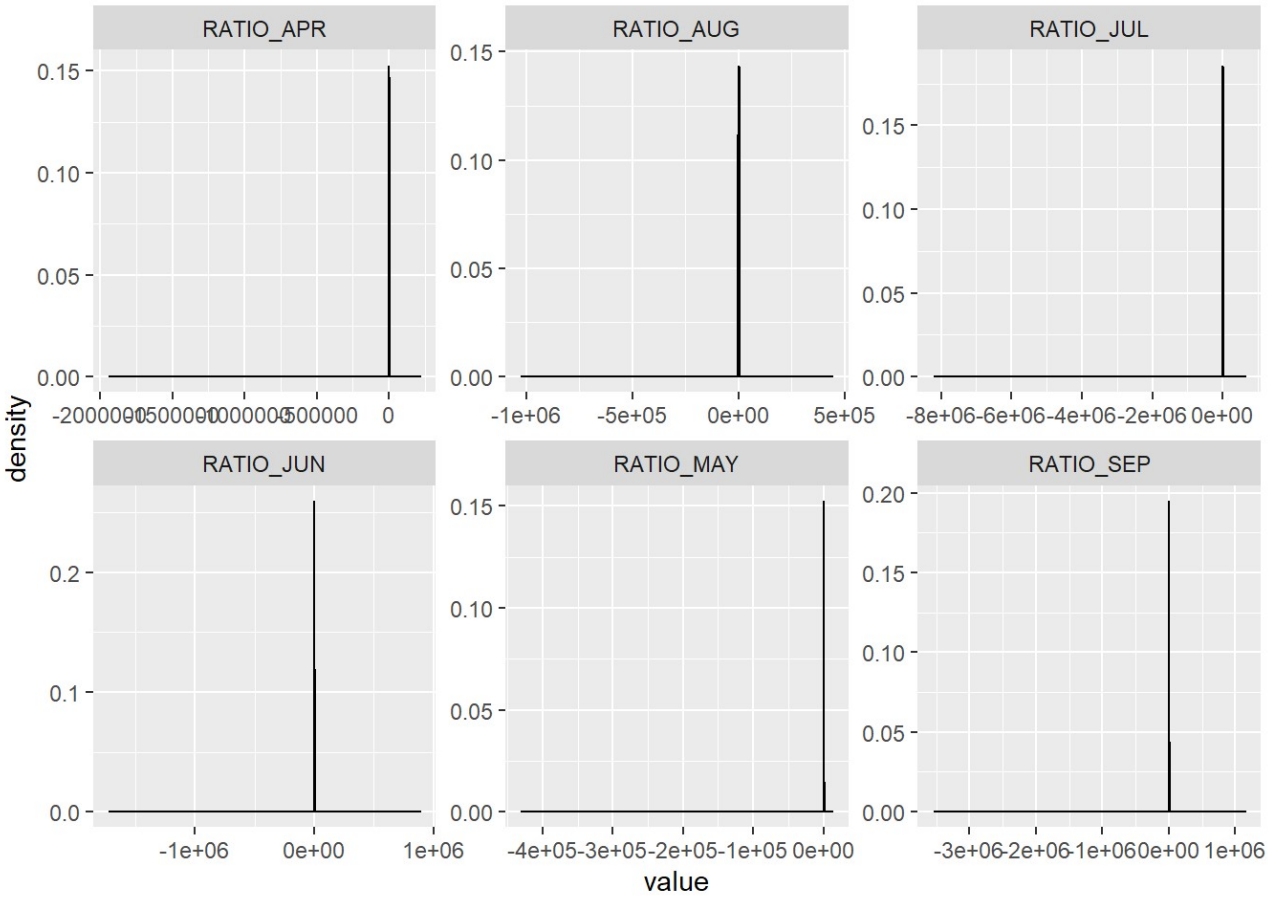



```
taiwan_bank3 <- subset (taiwan_bank, select = c(13:25))
taiwan_bank3$RATIO_SEP<- (taiwan_bank$PAID_AMT_SEP/taiwan_bank$BILL_AMT_SEP)*100
taiwan_bank3$RATIO_AUG<- (taiwan_bank$PAID_AMT_AUG/taiwan_bank$BILL_AMT_AUG)*100
taiwan_bank3$RATIO_JUL<- (taiwan_bank$PAID_AMT_JUL/taiwan_bank$BILL_AMT_JUL)*100
taiwan_bank3$RATIO_JUN<- (taiwan_bank$PAID_AMT_JUN/taiwan_bank$BILL_AMT_JUN)*100
taiwan_bank3$RATIO_MAY<- (taiwan_bank$PAID_AMT_JUN/taiwan_bank$BILL_AMT_MAY)*100
taiwan_bank3$RATIO_APR<- (taiwan_bank$PAID_AMT_JUN/taiwan_bank$BILL_AMT_APR)*100
ratio_paid <- (taiwan_bank3[,14:19])
sum(is.na(ratio_paid))
```

```
## [1] 14452
```

```
ratio_paid[is.na(ratio_paid)] <- 0
library(ggplot2)
 ratio_paid %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_density()
```

```
## Warning: Removed 3653 rows containing non-finite values (stat_density).
```



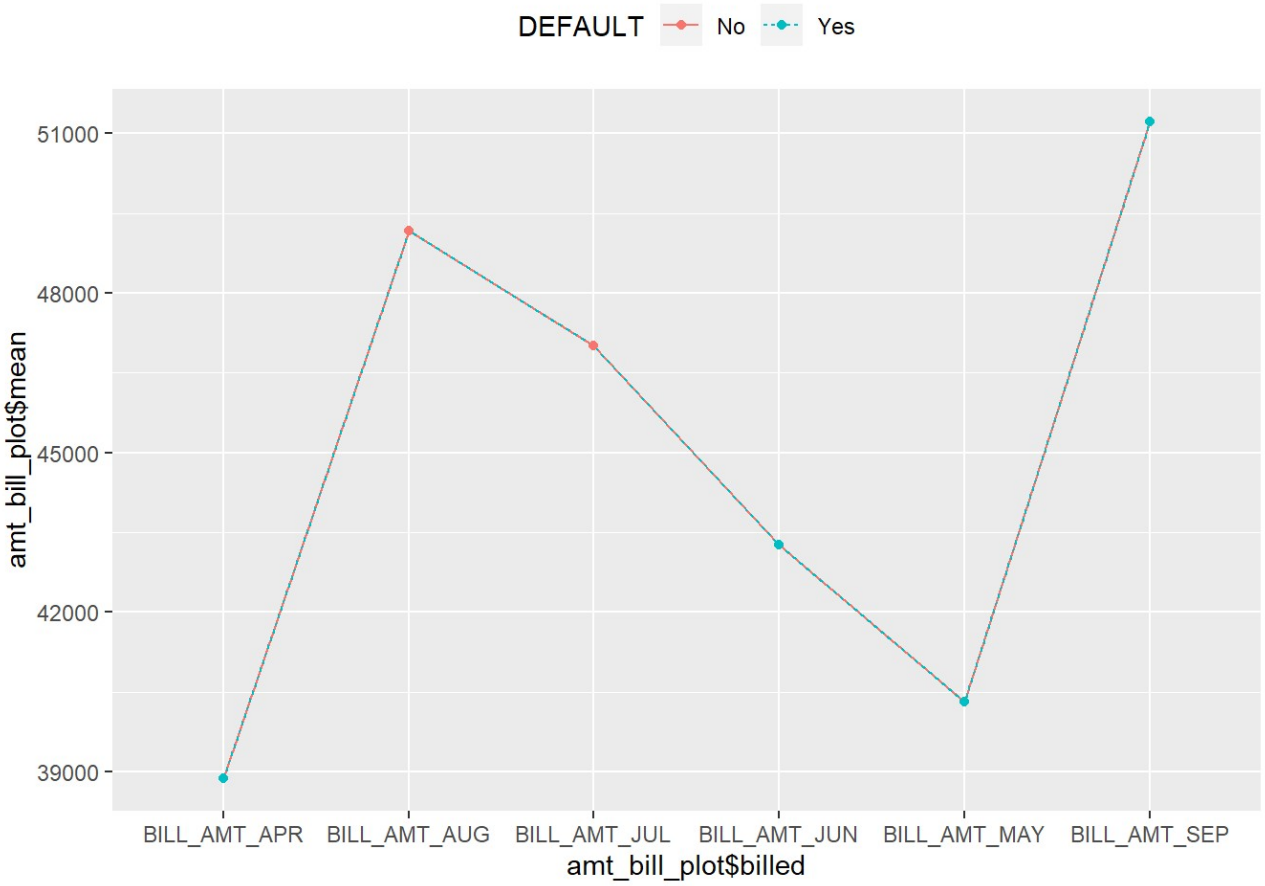

```
amt_bill<- taiwan_data %>%
    select(BILL_AMT_SEP,    BILL_AMT_AUG,   BILL_AMT_JUL,   BILL_AMT_JUN,   BILL_AMT_M
AY,   BILL_AMT_APR,) %>%
    psych::describe(quant=c(.25,.75)) %>%
    as_tibble(rownames="billed")  %>%
    print()
```

```
## # A tibble: 6 x 16
##   billed  vars     n   mean     sd median trimmed    mad     min    max
##   <chr>  <int> <dbl>  <dbl>  <dbl>  <dbl>   <dbl>  <dbl>   <dbl>  <dbl>
## 1 BILL_~     1 30000 51223. 73636. 22382.  35360. 32321. -165580 9.65e5
## 2 BILL_~     2 30000 49179. 71174. 21200   33836. 30853.  -69777 9.84e5
## 3 BILL_~     3 30000 47013. 69349. 20088.  32064. 29220. -157264 1.66e6
## 4 BILL_~     4 30000 43263. 64333. 19052   29212. 27659. -170000 8.92e5
## 5 BILL_~     5 30000 40311. 60797. 18104.  26921. 26225.  -81334 9.27e5
## 6 BILL_~     6 30000 38872. 59554. 17071   25726. 24841. -339603 9.62e5
## # ... with 6 more variables: range <dbl>, skew <dbl>, kurtosis <dbl>,
## #   se <dbl>, Q0.25 <dbl>, Q0.75 <dbl>
```

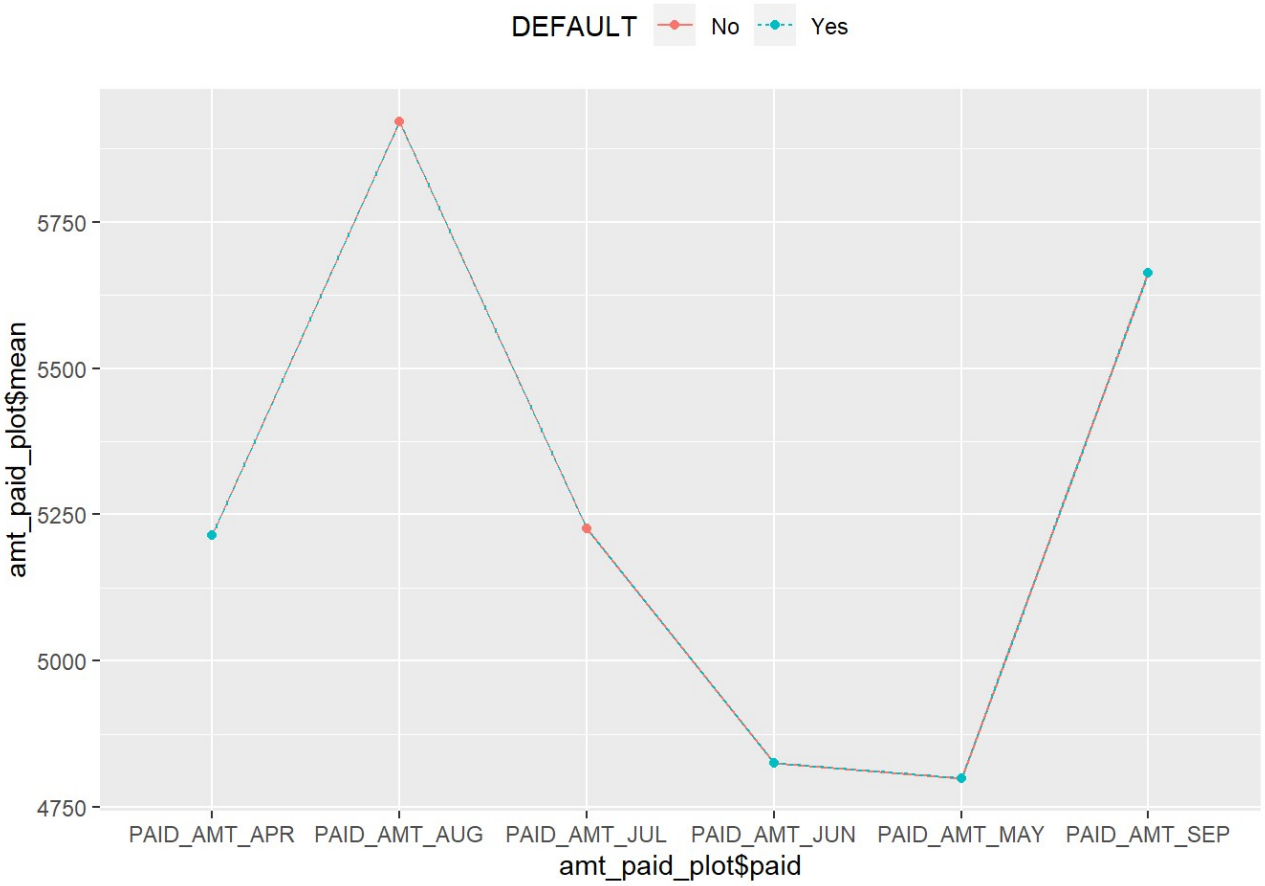
```
amt_paid<- taiwan_data %>%
    select(PAID_AMT_SEP,    PAID_AMT_AUG,   PAID_AMT_JUL,   PAID_AMT_JUN,   PAID_AMT_M
AY,   PAID_AMT_APR) %>%
    psych::describe(quant=c(.25,.75)) %>%
    as_tibble(rownames="paid")  %>%
    print()
```

```
## # A tibble: 6 x 16
##   paid   vars     n  mean     sd median trimmed   mad   min    max  range
##   <chr> <int> <dbl> <dbl>  <dbl>  <dbl>   <dbl> <dbl> <dbl>  <dbl>  <dbl>
## 1 PAID~     1 30000 5664. 16563.   2100   2997. 2864.     0 8.74e5 8.74e5
## 2 PAID~     2 30000 5921. 23041.   2009   2876. 2952.     0 1.68e6 1.68e6
## 3 PAID~     3 30000 5226. 17607.   1800   2469. 2661.     0 8.96e5 8.96e5
## 4 PAID~     4 30000 4826. 15666.   1500   2199. 2224.     0 6.21e5 6.21e5
## 5 PAID~     5 30000 4799. 15278.   1500   2202. 2224.     0 4.27e5 4.27e5
## 6 PAID~     6 30000 5216. 17777.   1500   2165. 2224.     0 5.29e5 5.29e5
## # ... with 5 more variables: skew <dbl>, kurtosis <dbl>, se <dbl>,
## #   Q0.25 <dbl>, Q0.75 <dbl>
```

```
amt_bill_plot = cbind(amt_bill,taiwan_bank$DEFAULT)
ggplot(amt_bill_plot, aes(x=amt_bill_plot$billed, y=amt_bill_plot$mean,color=DEFAULT,
group=DEFAULT)) +
  geom_line(aes(linetype=DEFAULT))+
  geom_point()+
  theme(legend.position="top")
```




```
amt_paid_plot = cbind(amt_paid,taiwan_bank$DEFAULT,taiwan_bank$SEX)
ggplot(amt_paid_plot, aes(x=amt_paid_plot$paid, y=amt_paid_plot$mean,color=DEFAULT, gr
oup=DEFAULT)) +
  geom_line(aes(linetype=DEFAULT))+
  geom_point()+
  theme(legend.position="top")
```

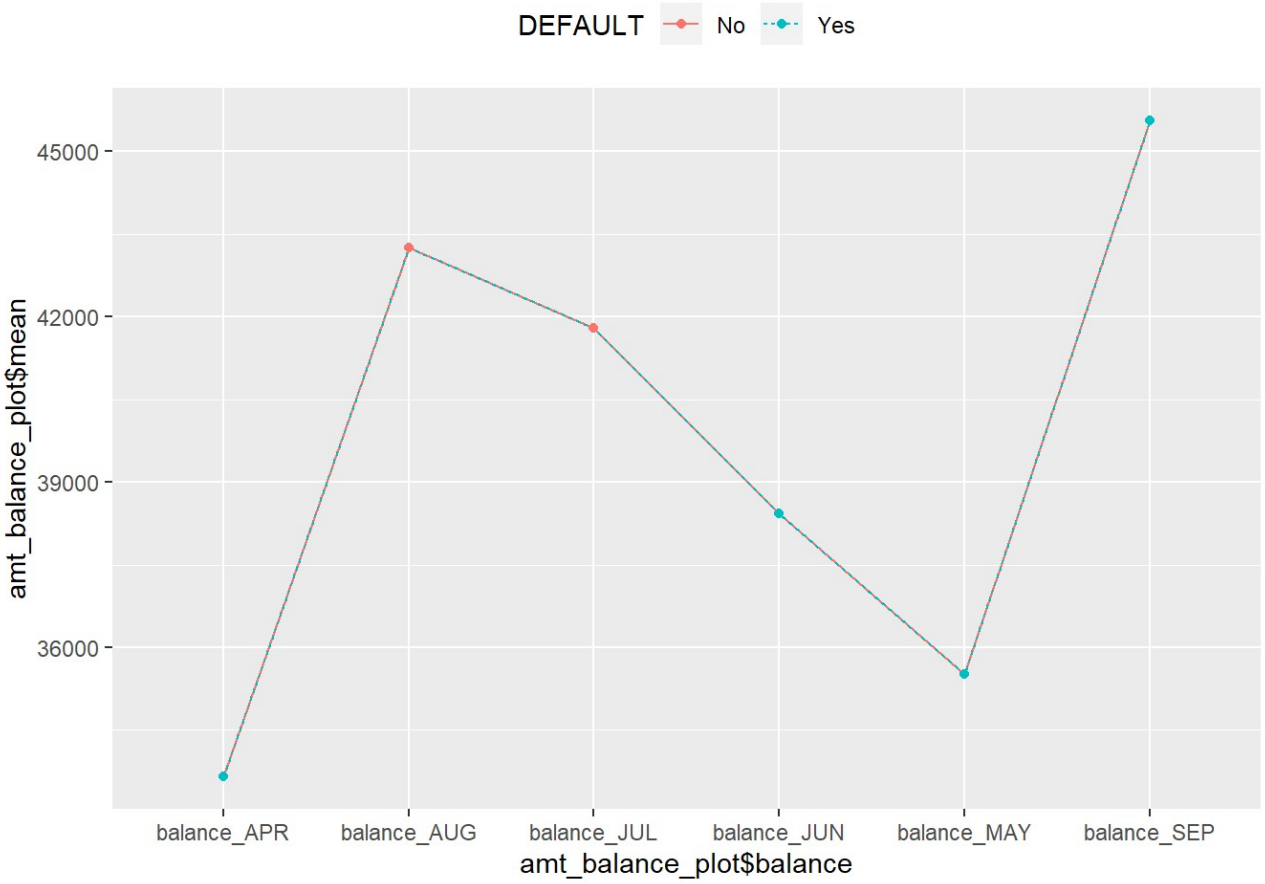



```
taiwan_bank3$balance_SEP<- (taiwan_data$BILL_AMT_SEP-taiwan_data$PAID_AMT_SEP)
taiwan_bank3$balance_AUG<- (taiwan_data$BILL_AMT_AUG-taiwan_data$PAID_AMT_AUG)
taiwan_bank3$balance_JUL<- (taiwan_data$BILL_AMT_JUL-taiwan_data$PAID_AMT_JUL)
taiwan_bank3$balance_JUN<- (taiwan_data$BILL_AMT_JUN-taiwan_data$PAID_AMT_JUN)
taiwan_bank3$balance_MAY<- (taiwan_data$BILL_AMT_MAY-taiwan_data$PAID_AMT_MAY)
taiwan_bank3$balance_APR<- (taiwan_data$BILL_AMT_APR-taiwan_data$PAID_AMT_APR)
```

```
amt_balance<- taiwan_bank3 %>%
    select(balance_SEP, balance_AUG,balance_JUL,balance_JUN,balance_MAY,balance_APR) %
>%
    psych::describe(quant=c(.25,.75)) %>%
    as_tibble(rownames="balance")  %>%
    print()
```

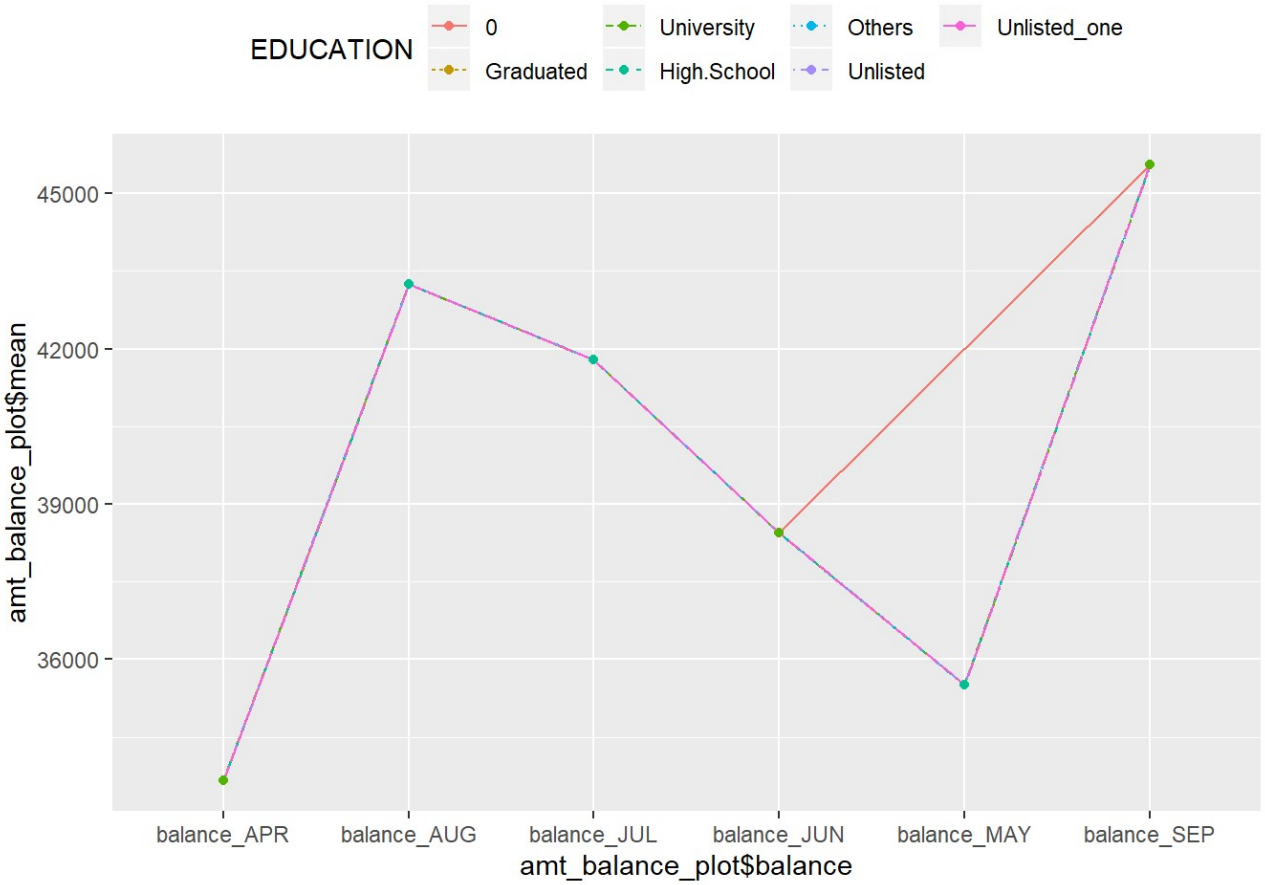
```
## # A tibble: 6 x 16
##   balance  vars     n   mean     sd median trimmed    mad     min    max
##   <chr>   <int> <dbl>  <dbl>  <dbl>  <dbl>   <dbl>  <dbl>   <dbl>  <dbl>
## 1 balanc~     1 30000 45560. 73174. 18550.  31662. 28043. -7.34e5 9.14e5
## 2 balanc~     2 30000 43258. 72566. 18102.  30248. 27328. -1.70e6 9.33e5
## 3 balanc~     3 30000 41787. 69295. 17769   28876. 26349. -8.55e5 1.54e6
## 4 balanc~     4 30000 38437. 64201. 16970   26314. 25160. -6.67e5 8.42e5
## 5 balanc~     5 30000 35512. 60553. 15538   23994. 23037. -4.14e5 8.77e5
## 6 balanc~     6 30000 33656. 60151. 13926.  22795. 20647. -6.85e5 9.11e5
## # ... with 6 more variables: range <dbl>, skew <dbl>, kurtosis <dbl>,
## #   se <dbl>, Q0.25 <dbl>, Q0.75 <dbl>
```

```
amt_balance_plot = cbind(amt_balance,taiwan_bank$DEFAULT,taiwan_bank$MARRIAGE,taiwan_b
ank$SEX,taiwan_bank$EDUCATION,taiwan_bank$LIMIT_BAL,taiwan_bank$AGE)
write.csv(amt_balance_plot, file = "amt_balance_plot.xls")
write.csv(taiwan_bank3, file = "taiwan_bank3.xls")
ggplot(amt_balance_plot, aes(x=amt_balance_plot$balance, y=amt_balance_plot$mean,colou
r=DEFAULT,group=DEFAULT)) +
  geom_line(aes(linetype=DEFAULT))+
  geom_point()+
  theme(legend.position="top")
```

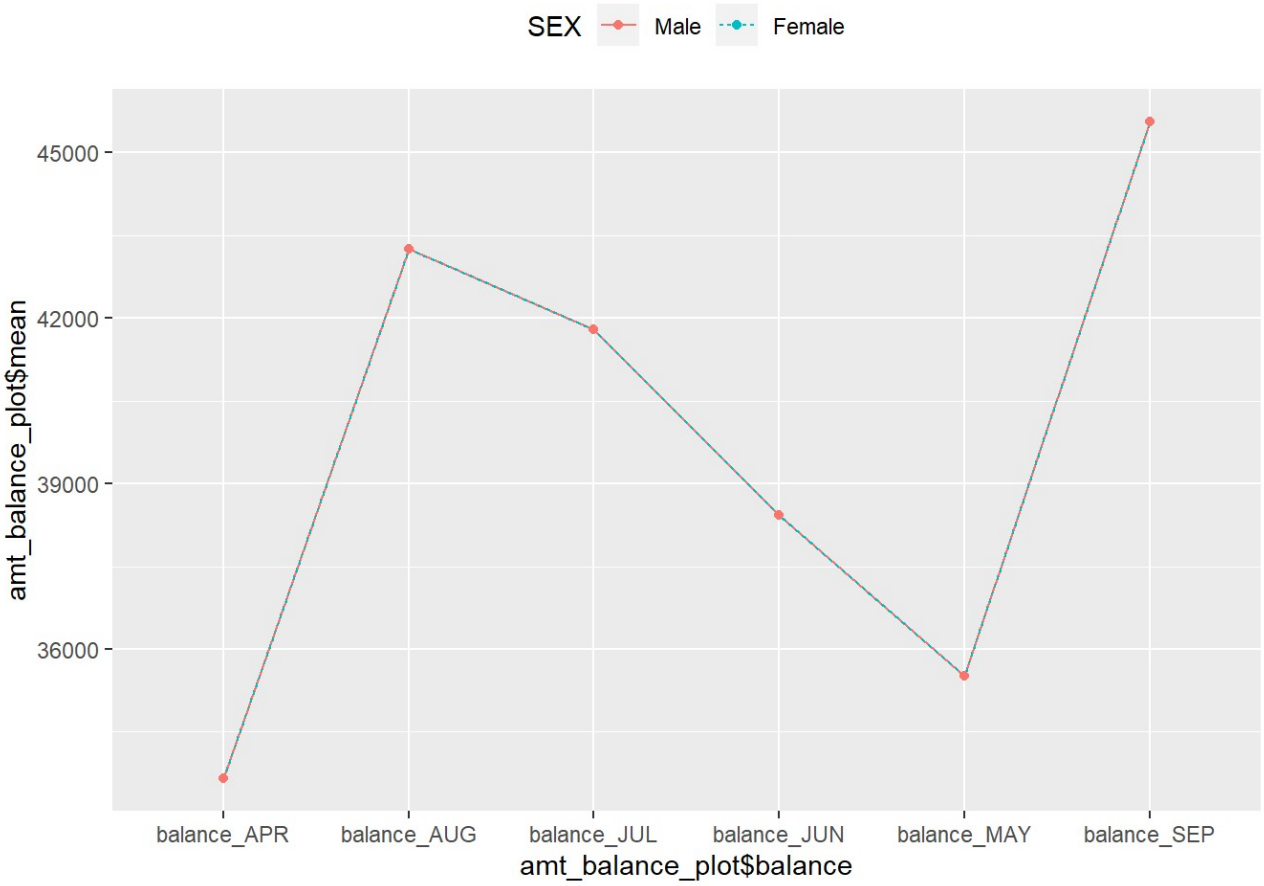


We found month of Aug,Jul and Sep 2005 has been shown the maximum due amount which is not paid. Now let us

```
ggplot(amt_balance_plot, aes(x=amt_balance_plot$balance, y=amt_balance_plot$mean, colo
r=EDUCATION,group=EDUCATION)) +
  geom_line(aes(linetype=EDUCATION))+
  geom_point()+
  theme(legend.position="top")
```




```
ggplot(amt_balance_plot, aes(x=amt_balance_plot$balance, y=amt_balance_plot$mean,color
=SEX, group=SEX)) +
  geom_line(aes(linetype=SEX))+
  geom_point()+
  theme(legend.position="top")
```


SEX
Male
Female
45000
42000
39000
36000
amt_balance_plot$mean
balance_APR
balance_AUG
balance_JUL
balance_JUN
balance_MAY
balance_SEP
amt_balance_plot$balance