Loan Interest Rates

Ke Guan

6/30/2019

This is a markdown document of determining interest rates for loans.

```
load_data=read.csv(file="C:/Users/nimok/Desktop/Analyst_Test/loan_interest_ra
tes.csv", header=T,na.strings=c("","NA"))
```

1. Explore variables of this dataset

1.1 Type Checking, fix and imputation

```
#take a quick look
str(load_data)

## 'data.frame':    400000 obs. of  27 variables:
##  $ X1 : Factor w/ 482 levels "10.00%","10.01%",..: 58 22 234 97 113 291 3
140 442 171 ...
##  $ X2 : int  54734 55742 57167 57245 57416 58524 58915 59006 61390 61419 .
..
##  $ X3 : int  80364 114426 137225 138150 139635 149512 153417 154254 182594
 182917 ...
##  $ X4 : Factor w/ 1339 levels "$1,000 ","$1,025 ",..: 681 1214 681 9 73 12
22 1234 878 1093 1157 ...
##  $ X5 : Factor w/ 1342 levels "$1,000 ","$1,025 ",..: 681 1217 681 9 73 12
25 1137 879 1096 1160 ...
##  $ X6 : Factor w/ 7036 levels "$0 ","$1,000 ",..: 2393 5874 3667 44 344 59
42 5525 2894 4367 5322 ...
##  $ X7 : Factor w/ 2 levels " 36 months"," 60 months": 1 1 1 1 1 1 1 1 1 1
...
##  $ X8 : Factor w/ 7 levels "A","B","C","D",..: 2 2 4 3 3 4 2 3 1 4 ...
##  $ X9 : Factor w/ 35 levels "A1","A2","A3",..: 9 10 18 12 13 19 8 15 5 17
...
##  $ X10: Factor w/ 187823 levels "'roduction manager",..: NA 35290 183252 3
1973 158969 10666 132686 NA NA 123534 ...
##  $ X11: Factor w/ 12 levels "< 1 year","1 year",..: 1 1 2 3 8 11 5 5 1 2 .
..
##  $ X12: Factor w/ 6 levels "ANY","MORTGAGE",..: 6 6 6 5 6 6 6 2 2 6 ...
##  $ X13: num  85000 65000 70000 54000 32000 58000 85000 80800 148000 45000
...
##  $ X14: Factor w/ 3 levels "not verified",..: 2 1 2 1 1 3 1 1 1 1 ...
##  $ X15: Factor w/ 91 levels "10-Apr","10-Aug",..: 81 76 50 8 89 26 68 91 4
 4 ...
##  $ X16: Factor w/ 122043 levels "- Pay off Dell Financial: $ 1300.00 - Pay
 off IRS for 2005: $ 1400.00 - Pay off Mac Comp    : $ 1700.00 - Pay o"| __tr
uncated__,..: 117568 120194 121735 120111 118935 117471 117874 118787 118749
```

```
121389 ...
##  $ X17: Factor w/ 14 levels "car", "credit card",..: 3 2 3 3 3 3 3 2 2 3 .
..
##  $ X18: Factor w/ 61626 levels "'08 & '09 Roth IRA Investments",..: 19148
14809 39556 61626 43259 6872 54545 52589 33039 3435 ...
##  $ X19: Factor w/ 50 levels "AK","AL","AR",..: 5 34 34 43 7 39 5 43 43 21
...
##  $ X20: num  19.48 14.29 10.5 5.47 11.63 ...
##  $ X21: int  0 0 0 0 0 0 0 1 0 0 ...
##  $ X22: Factor w/ 660 levels "1-Apr","1-Aug",..: 293 572 394 336 254 165 5
24 392 517 7 ...
##  $ X23: int  NA NA 41 64 58 26 NA 13 NA 38 ...
##  $ X24: int  NA NA NA NA NA NA NA 0 NA 63 ...
##  $ X25: int  0 0 0 0 0 0 0 0 0 1 ...
##  $ X26: int  42 7 17 31 40 25 11 23 19 9 ...
##  $ X27: Factor w/ 2 levels "f","w": 1 1 1 1 1 1 1 1 1 1 ...
```

X1

X1 is the interest rate on the loan which is the prediction. According to the rate information posted by the Lending Club, the interest rates take credit risk and market conditions into account. The final interest rates are influenced by the loan grades modified to the Base Risk Subgrades. The Lending Club utilizes credit risk indicators including request loan amount and loan maturity to modify these Subgrades. X1 contains the percentages. To better predict the interest rates, need to extract the numbers from X1 which is stored as a categorical factor. And omit observations with NA in X1.

```
load_data$X1=as.numeric(gsub("%","", load_data$X1))
typeof(load_data$X1)

## [1] "double"

sum(is.na(load_data$X1))

## [1] 61010

load_data1=load_data[!is.na(load_data$X1),]
dim(load_data1)

## [1] 338990     27
```

X2 and X3

X2 and X3 are unique ids for the loan and borrowers, and they don't contribute too considerably to the prediction. Therefore, these two columns can be dropped. Before dropping these two columns, check if there are duplicate ids for X2. If any two ids are the same, the duplicate observation will be deleted. The result shows there is no duplicate observation.

```
n_occur=data.frame(table(load_data1$X2))
n_occur[n_occur$Freq>1,]
```

```
## [1] Var1 Freq
## <0 rows> (or 0-length row.names)

rm(n_occur)
# drop X2, X3
load_data1=within(load_data1, rm(X2, X3))
dim(load_data1)

## [1] 338990     25
```

X4, X5, X6

X4(the loan amount requested), as mentioned hitherto, is an indicator for Base Risk Subgrade which comprises a significant factor based on the risk information of the Lending Club. From the above glimpse of the data, X4 is categorical with 1340 levels. First, let's see if there exists missing value.

```
#na value
sum(is.na(load_data1$X4))

## [1] 1
```

Only one observation has NA in X4. Take a close look at this observation.

```
load_data1[is.na(load_data1$X4),]

##          X1   X4   X5   X6   X7   X8   X9  X10  X11  X12 X13  X14  X15
## 364112 7.69 <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>  NA <NA> <NA>
##         X16  X17  X18  X19 X20 X21  X22 X23 X24 X25 X26  X27
## 364112 <NA> <NA> <NA> <NA>  NA  NA <NA>  NA  NA  NA  NA <NA>
```

This observation merely has value in X1, then we can simply drop this observation.

```
load_data1=load_data1[!is.na(load_data1$X4),]

dim(load_data1)

## [1] 338989     25
```

As it includes thousands of levels with numeric values, considering it as a numeric variable would be more meaningful. X5, the loan amount funded, X6, an investor-funded portion of the loan, are also categorical with thousands of levels. So, converting them to numbers without "$" as well.

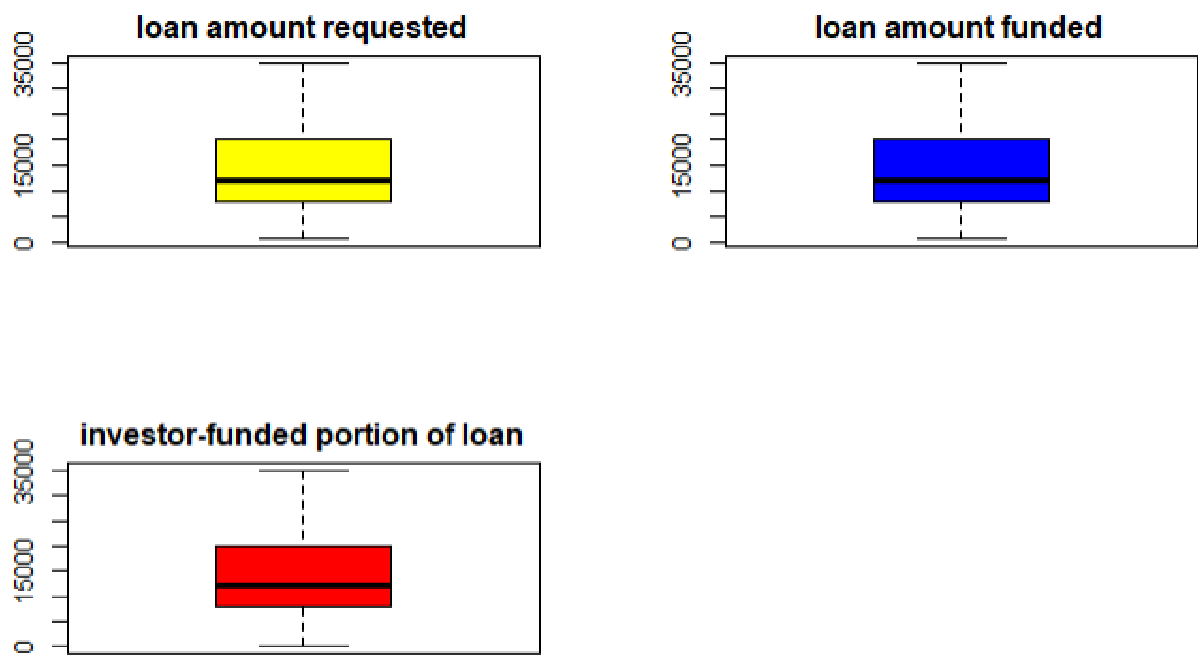
```
sum(is.na(load_data1$X5))

## [1] 0

sum(is.na(load_data1$X6))

## [1] 0
```

```
load_data1$X4=as.numeric(gsub(",|\\$","", load_data1$X4))
load_data1$X5=as.numeric(gsub(",|\\$","", load_data1$X5))
load_data1$X6=as.numeric(gsub(",|\\$","", load_data1$X6))
head(load_data1)
```

```
##      X1    X4    X5    X6         X7 X8 X9                   X10
## 1 11.89 25000 25000 19080  36 months  B B4                  <NA>
## 2 10.71  7000  7000   673  36 months  B B5                   CNN
## 3 16.99 25000 25000 24725  36 months  D D3         Web Programmer
## 4 13.11  1200  1200  1200  36 months  C C2 city of beaumont texas
## 5 13.57 10800 10800 10692  36 months  C C3   State Farm Insurance
## 6 19.05  7200  7200  7200  36 months  D D4             Arkwright
##         X11  X12   X13                       X14    X15
## 1  < 1 year RENT 85000        VERIFIED - income  9-Aug
## 2  < 1 year RENT 65000             not verified  8-May
## 3    1 year RENT 70000        VERIFIED - income 14-Aug
## 4 10+ years  OWN 54000             not verified 10-Mar
## 5   6 years RENT 32000             not verified  9-Nov
## 6   9 years RENT 58000 VERIFIED - income source 12-Aug
##  X16
## 1 Due to a lack of personal finance education and exposure to poor financi
ng skills growing up, I was easy prey for credit predators. I am devoted to b
ecoming debt-free and can assure my lenders that I will pay on-time every tim
e. I have never missed a payment during the last 16 years that I have had cre
dit.
## 2 Just want to pay off the last bit of credit card debt at a better rate.
## 3Trying to pay a friend back for apartment broker's fee incurred from as w
ell as credit card stuff.
## 4 If funded, I would use this loan consolidate two loans with interest rat
es of 15 and 16 percent respectively.   I have no mortgage.  One car is paid
for and the other I bought from my sister.  I pay her $200 / month.  I owe he
r about $1000.  The biggest monthly expense we have is tuition for two kids g
oing to Catholic School, ($600 / month).    I have been on the same job since
 1990, with a salary of $54,000.  My husband has been on the same job since 1
995, with a salary of $30,000.  My monthly expenses run about $2750.    Borro
wer added on 03/11/10 > We have really worked hard to clean up our credit dur
ing the past five years.  We are really wanting to use this loan to continue
that by paying off higher interest loans with this loan.<br/>
## 5I currently have a personal loan with Citifinancial that I have a high in
terest rate on I need 7000 to pay this off.  I also have 3 other creidit card
s I would like to pay off with this loan to get this into one easy payment.
139635 added on 11/04/09 > Having one monthly payment will be a lot easier in
stead of making multiple payments to different companies.  I have paid all of
 my bills on time<br/>
## 6 Credit cards are out of here, I am tired of being in debt.  I have a gir
lfriend and her daughter, that I am starting a life with.  I will pay all of
this off one way or another, but I would rather do it with your help instead
of the faceless bank!  Can you help me?
## X17  X18 X19   X20 X21
## 1 debt_consolidation Debt consolidation for on-time payer  CA 19.48   0
```

```
## 2         credit_card                            Credit Card payoff  NY 14.29   0
## 3 debt_consolidation                                           mlue  NY 10.50   0
## 4 debt_consolidation                                          zxcvb  TX  5.47   0
## 5 debt_consolidation                                   Nicolechr1978  CT 11.63   0
## 6 debt_consolidation                                      caminijio  RI  2.05   0
##      X22 X23 X24 X25 X26 X27
## 1 Feb-94  NA  NA   0  42   f
## 2 Oct-00  NA  NA   0   7   f
## 3 Jun-00  41  NA   0  17   f
## 4 Jan-85  64  NA   0  31   f
## 5 Dec-96  58  NA   0  40   f
## 6 Apr-94  26  NA   0  25   f
```

```
par(mfrow=c(2,2))
boxplot(load_data1$X4, main='loan amount requested', col='yellow')
boxplot(load_data1$X5, main="loan amount funded", col='blue')
boxplot(load_data1$X6, main='investor-funded portion of loan', col='red')
par(mfrow=c(1,1))
```



X7, X8, X9

X7, the number of payments (36 or 60), is the other significant indicator to modify the Base Risk Subgrade. Take a look at the distribution of diverse levels.

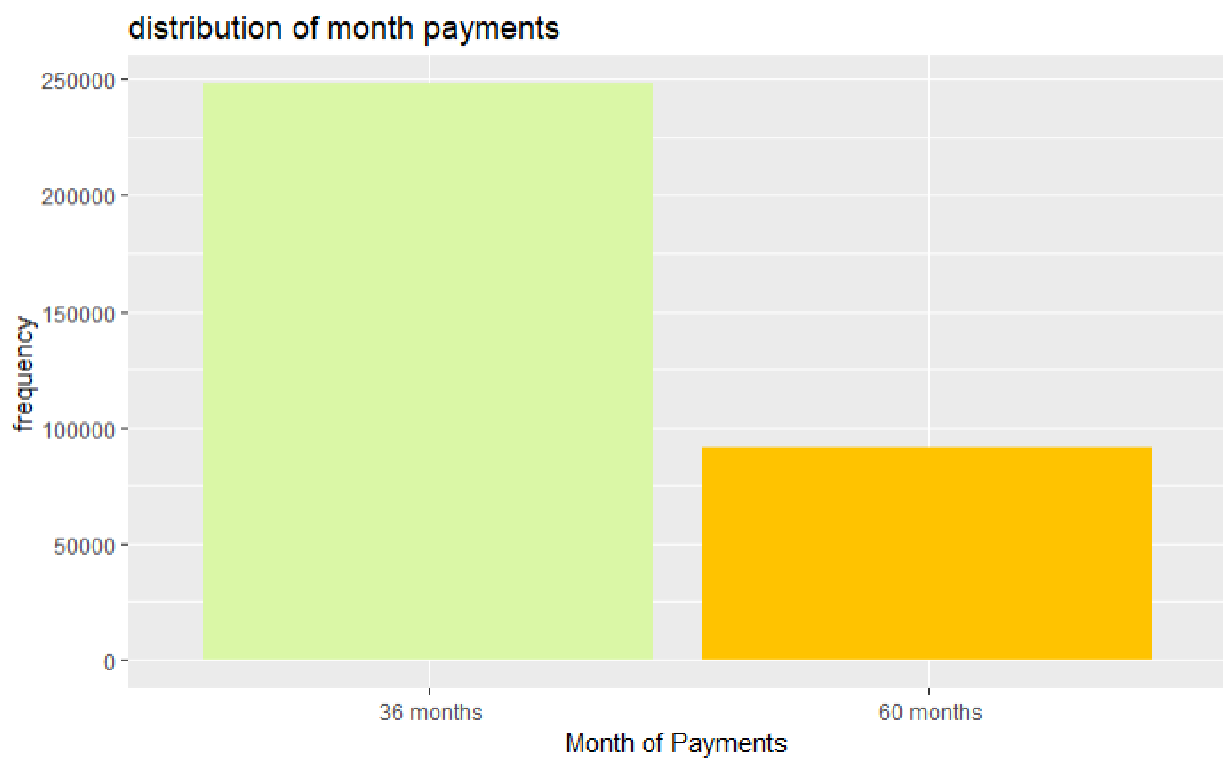
```
# distribution of the categorical variable
sum(is.na(load_data1$X7))

## [1] 0

levels(load_data1$X7)
```

```
## [1] " 36 months" " 60 months"
```

```
# drop the unused level
load_data1$X7=droplevels(load_data1)$X7
x7_freq=table(load_data1$X7)
x7_freq
```

```
##
##  36 months  60 months
##     247791      91198
```

```
library(ggplot2)
#barplot(x7_freq, xlab='number of payments', width=0.1, col='yellow',main="di
stribution for number of payment", ylab="Frequency")
ggplot(load_data1, aes(x=X7))+geom_bar(stat = 'count', fill=c("#DAF7A6","#FFC
300"))+labs(fill="month of payments",x="Month of Payments", y="frequency")+gg
title("distribution of month payments ")
```



X8 and X9, the loan grade and subgrade are the leading factors to evaluate the interest rate on the loan. There is a sizeable portion of missing value in X9, as well as X8

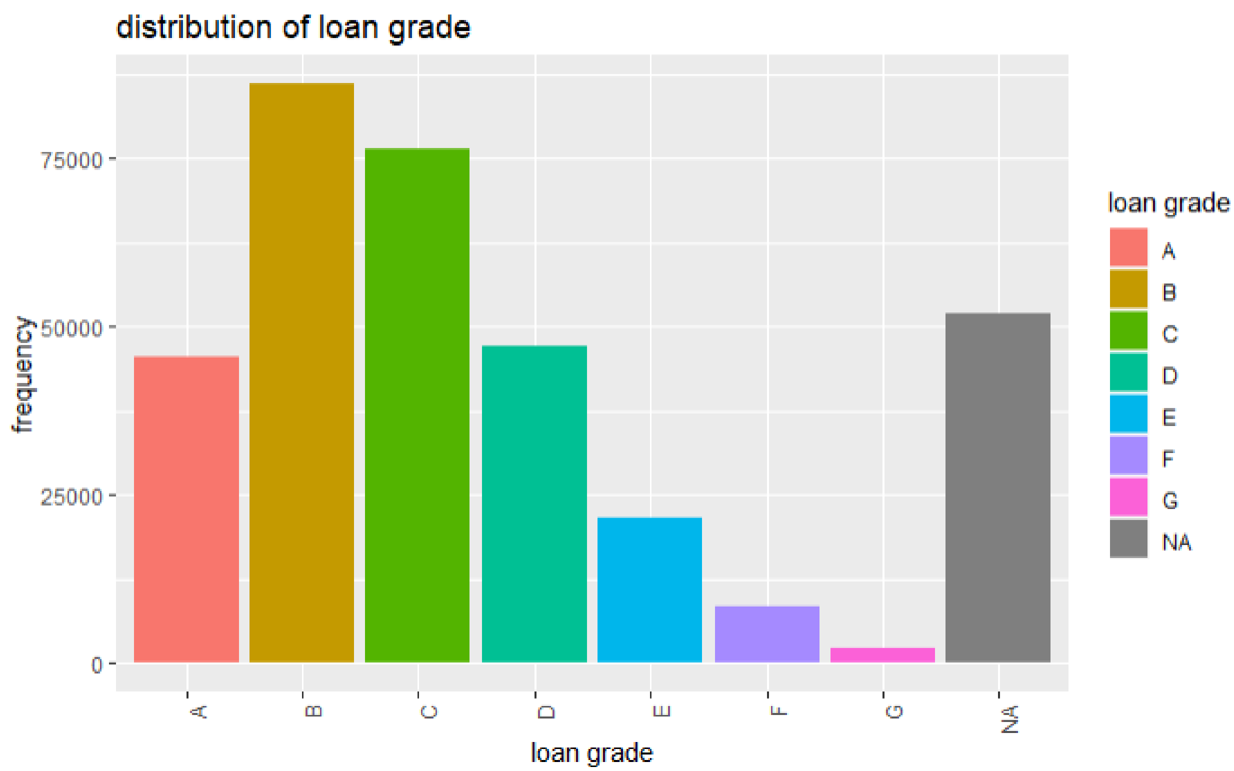
```
sum(is.na(load_data1$X8))
```

```
## [1] 51866
```

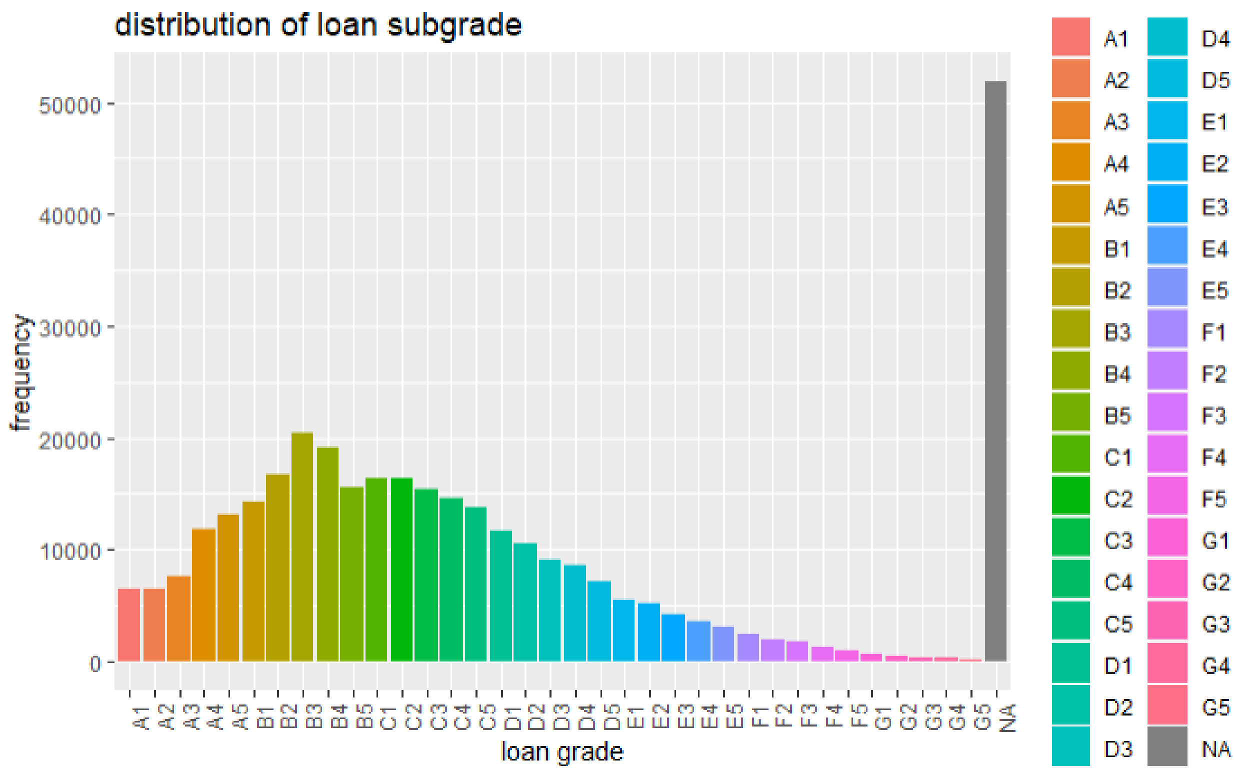
```
sum(is.na(load_data1$X9))
```

```
## [1] 51866
```

```r
load_data1$X8=droplevels(load_data1)$X8
x8_freq=table(load_data1$X8)
x8_prop=prop.table(x8_freq)
load_data1$X9=droplevels(load_data1)$X9
x9_freq=table(load_data1$X9)
x9_prop=prop.table(x9_freq)
## distribution of the grade and subgrade
ggplot(load_data1, aes(x=X8))+geom_bar(stat = 'count', aes(fill=X8))+theme(ax
is.text.x = element_text(angle = 90, hjust = 1))+labs(x="loan grade",fill="lo
an grade", y="frequency")+ggtitle("distribution of loan grade ")
```




```r
ggplot(load_data1, aes(x=X9))+geom_bar(stat = 'count',aes(fill=X9))+theme(axi
s.text.x = element_text(angle = 90, hjust = 1))+labs(fill="loan grade", y="fr
equency")+ggtitle("distribution of loan subgrade ")
```



Indicators for modifying subgrade are X4 (loan amount request) and X7 (number of payment). First, divide the data into two parts. One with missing X9 values, another part without.

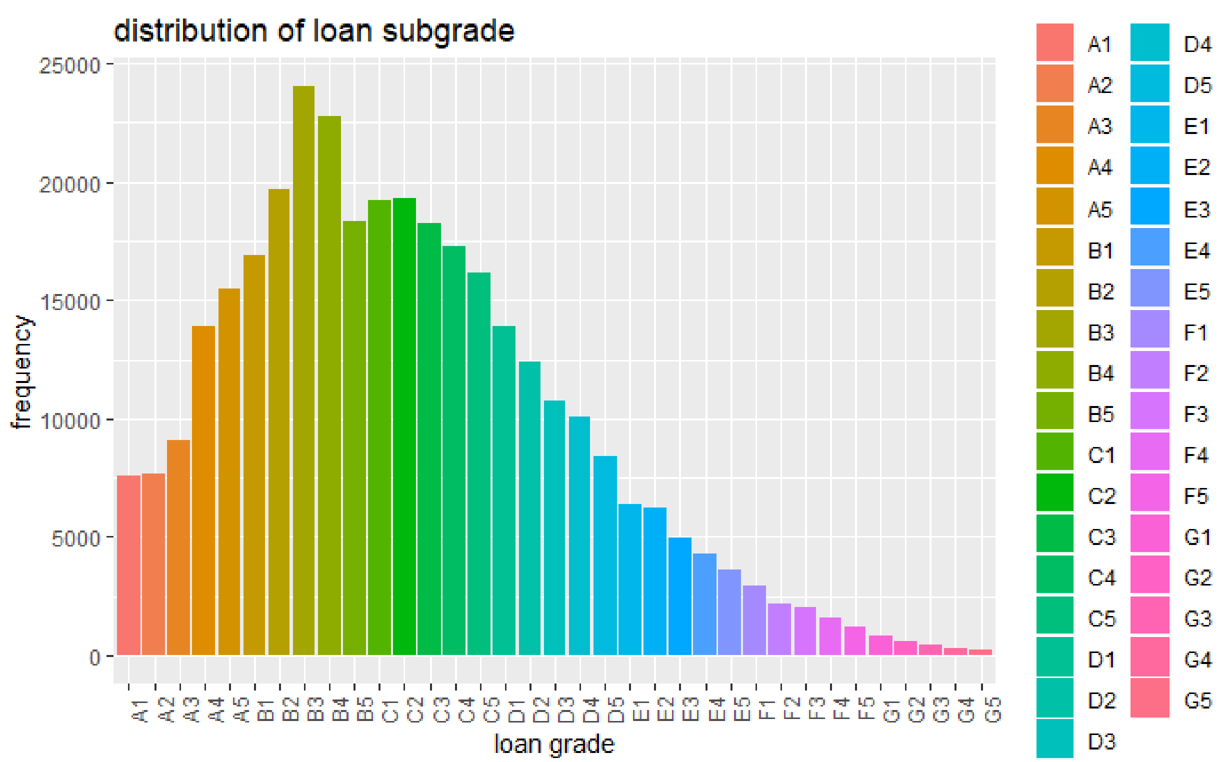
```
train=load_data1[!is.na(load_data1$X9),]
test=load_data1[is.na(load_data1$X9),]
```

To detect the missing of X9 is missing at random (MAR) or not. From the result, p=0.3963 represents the true difference in means is 0. Hence, consider the X9 as missing at Random.

```
t.test(train$X4, test$X4)

##
##  Welch Two Sample t-test
##
## data:  train$X4 and test$X4
## t = -0.84826, df = 72064, p-value = 0.3963
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -110.22082   43.63438
## sample estimates:
## mean of x mean of y
##  14271.87  14305.17
```

Using sequential hot-deck imputation to fill the missing data as X9 containing missing value is sorted according to one or more auxiliary variables.

```
x=load_data1$X9
tail(x, n=1)

## [1] C2
## 35 Levels: A1 A2 A3 A4 A5 B1 B2 B3 B4 B5 C1 C2 C3 C4 C5 D1 D2 D3 D4 ... G5

#last value is not empty
seqImpute=function(x){
  n=length(x)
  i=is.na(x)
  while(any(i)){
    x[i]=x[which(i)+1]
    i=is.na(x)
  }
  x[1:n]
}
## distribution of subgrade after filling the missing data
load_data1$X9=seqImpute(x)
load_data1$X9=droplevels(load_data1)$X9
x9_freq=table(load_data1$X9)
x9_prop=prop.table(x9_freq)
ggplot(load_data1, aes(x=X9))+geom_bar(stat = 'count',aes(fill=X9))+theme(axi
s.text.x = element_text(angle = 90, hjust = 1))+labs(fill="loan grade", y="fr
equency")+ggtitle("distribution of loan subgrade ")
```

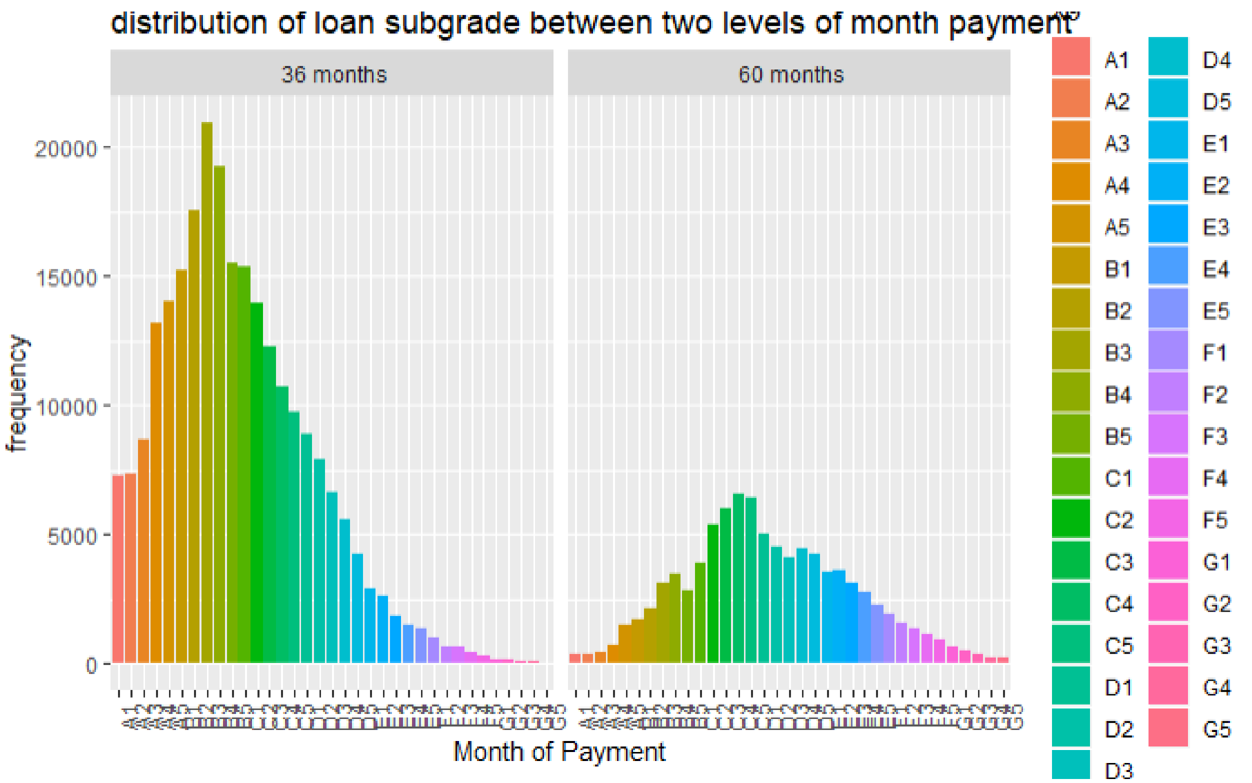


X8, the loan grade, can be extracted from X9.

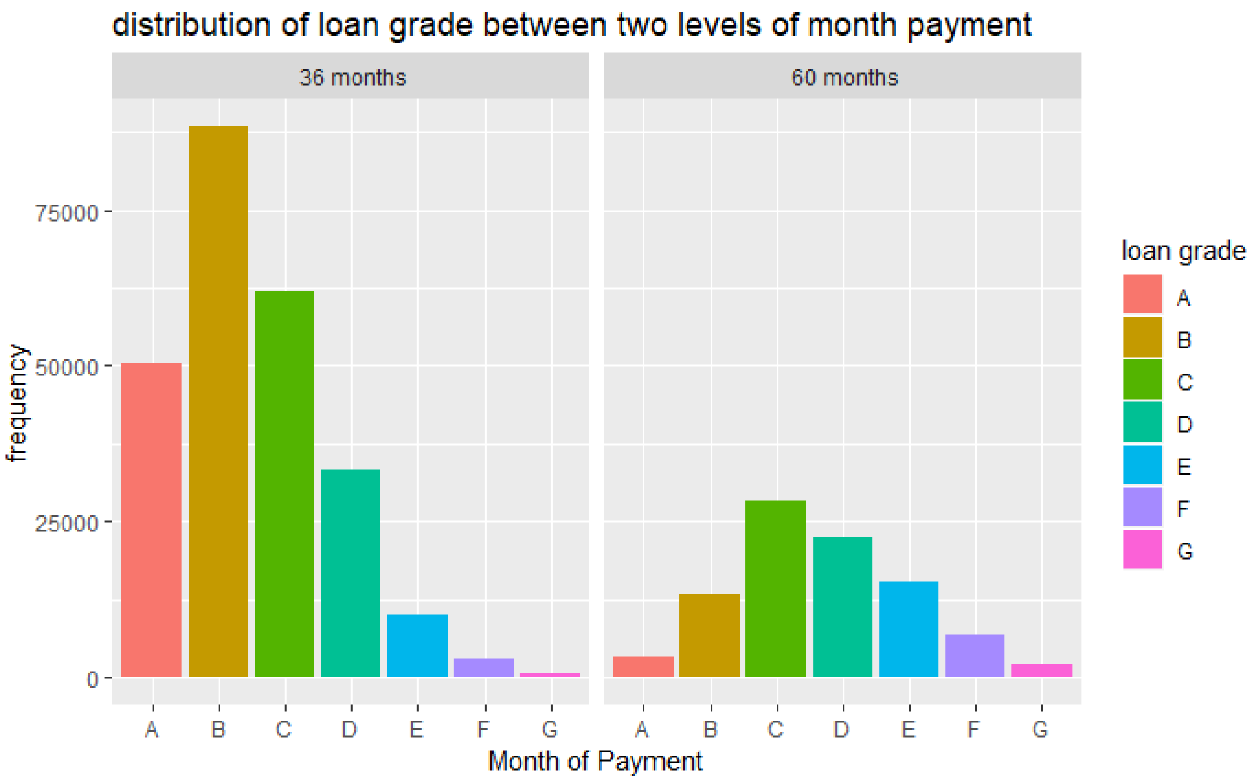
```
load_data1$X8=as.factor(gsub("[^a-zA-Z]", "",load_data1$X9 ))
## check the missing value
sum(is.na(load_data1$X8))

## [1] 0

sum(is.na(load_data1$X9))

## [1] 0
```

Let's identify the relationship between X9 and X7. Compared to 36 months, borrowers prefer 36 months payments.

```
library(ggplot2)
ggplot(load_data1, aes(x=X9))+geom_bar(stat = 'count',aes(fill=X9))+facet_gri
d(~X7)+theme(axis.text.x = element_text(angle = 90, hjust = 1))+labs(fill="lo
an grade",x="Month of Payment", y="frequency")+ggtitle("distribution of loan
subgrade between two levels of month payment")
```



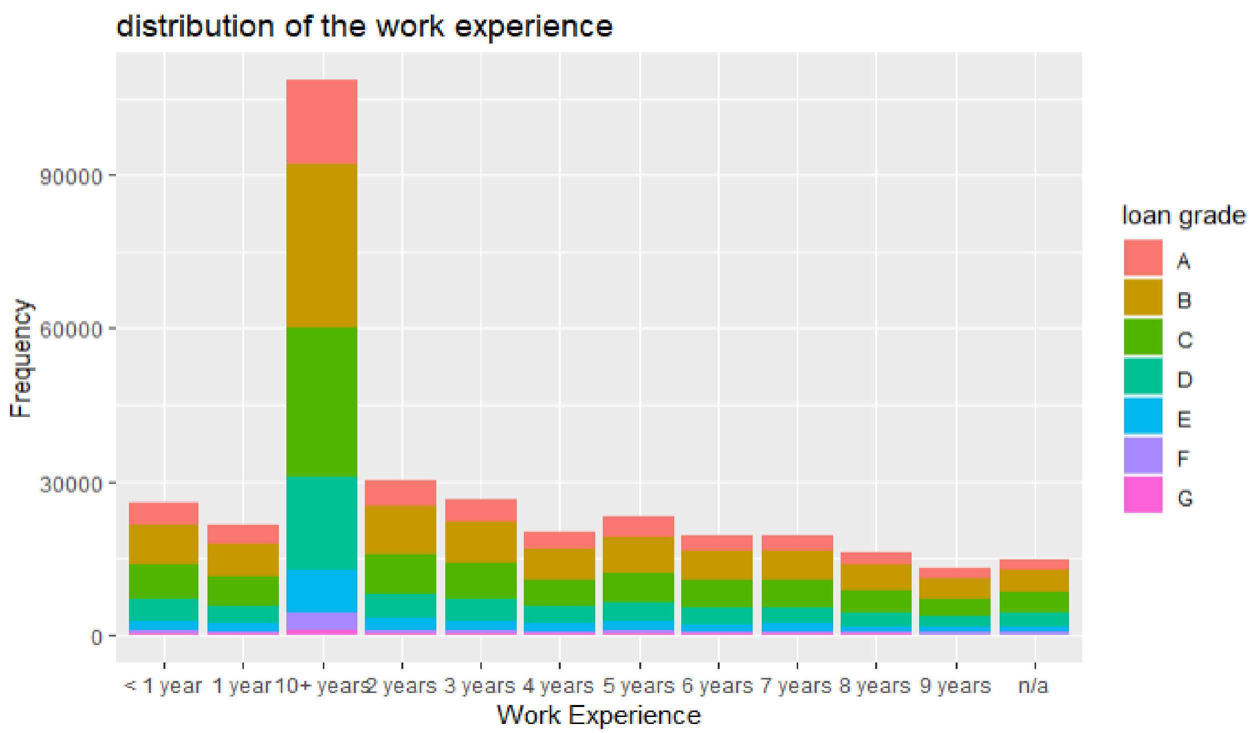

```
ggplot(load_data1, aes(x=X8))+geom_bar(stat = 'count',aes(fill=X8))+facet_gri
d(~X7)+labs(fill="loan grade",x="Month of Payment", y="frequency")+ggtitle("d
istribution of loan grade between two levels of month payment")
```



X10, X11

X10 (self-filled employer of job title) shows 187823 different values and 23969 missing values. X11, number of work experience (0 to 10; 10=10 or more), possesses 12 levels. Its distribution indicates the number of borrowers with 10+ work experience about 2-3 times more than others.

```
sum(is.na(load_data1$X10))

## [1] 20241

sum(is.na(load_data1$X11))

## [1] 0

load_data1$X11=droplevels(load_data1)$X11
ggplot(load_data1, aes(x=X11))+geom_bar(stat = "count", aes(fill=X8))+labs(fi
ll="loan grade", x="Work Experience", y="Frequency")+ggtitle("distribution of
 the work experience")
```

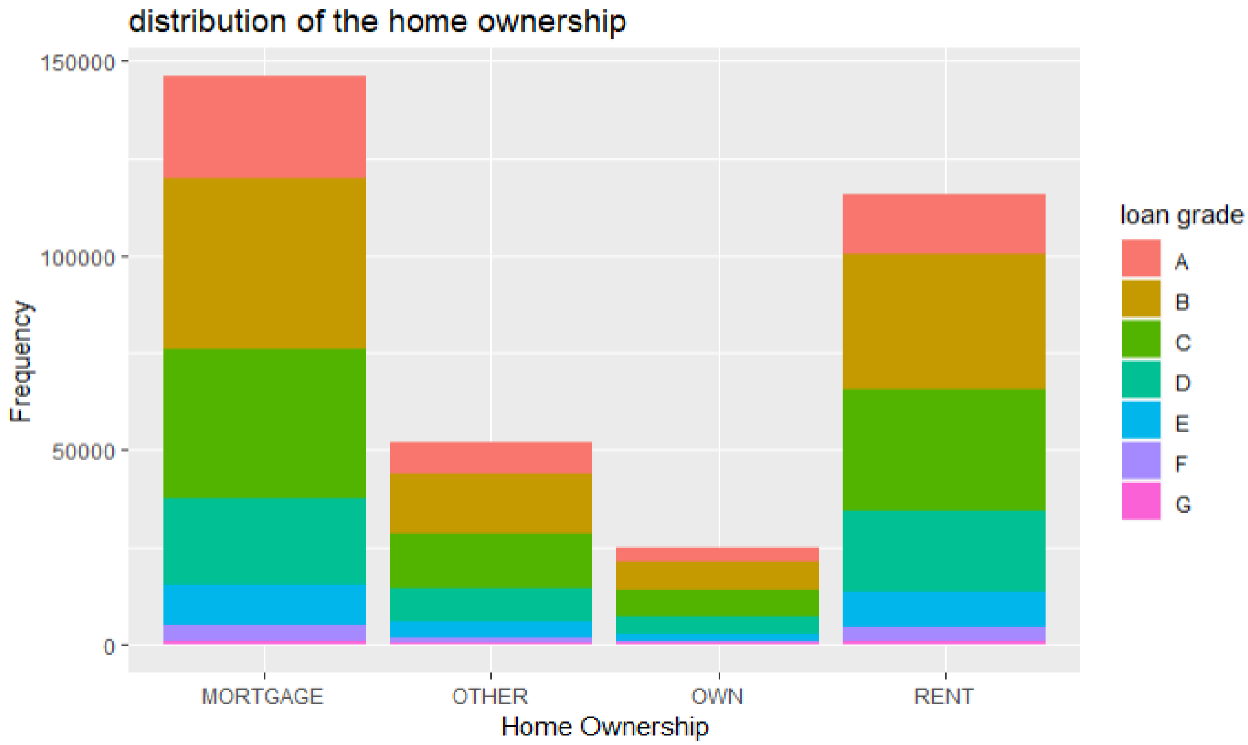


X12

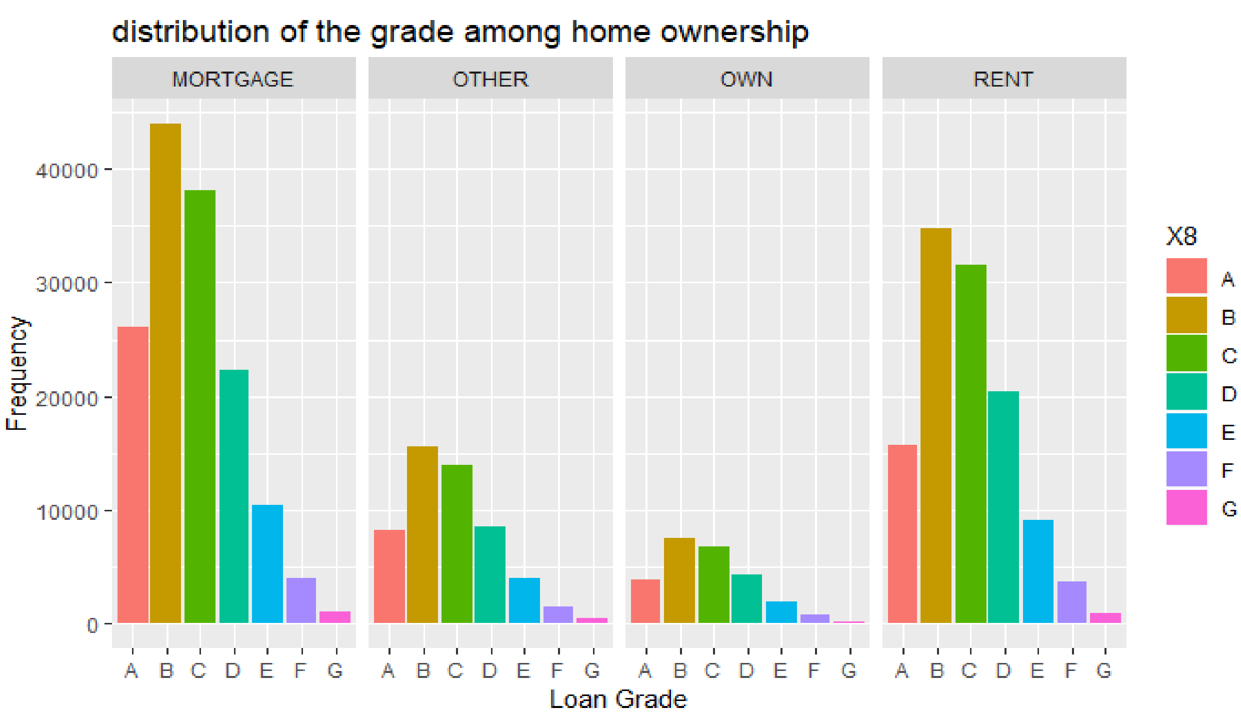
X12, home ownership status, an indicator reflects the ability to pay off the loan. Fill the missing values of house ownership status with "UNKNOWN'. And combine "ANY", "OTHER", and "NONE" together to reset the level to "OTHER".

```
library(car)

## Loading required package: carData

sum(is.na(load_data1$X12))

## [1] 51959

load_data1$X12=droplevels(load_data1)$X12
x12_prop=prop.table(table(load_data1$X12))
x12_prop

##
##          ANY     MORTGAGE         NONE        OTHER          OWN
## 3.483956e-06 5.085113e-01 1.045187e-04 3.727833e-04 8.701529e-02
##         RENT
## 4.039926e-01

load_data1$X12_cb=Recode(load_data1$X12, "c(NA, 'ANY', 'NONE','OTHER')='OTHER
'")
x12_prop=prop.table(table(load_data1$X12_cb))
ggplot(load_data1, aes(x=X12_cb))+geom_bar(stat = "count", aes(fill=X8))+labs
```

```
(fill="loan grade", x="Home Ownership", y="Frequency")+ggtitle("distribution
of the home ownership")
```



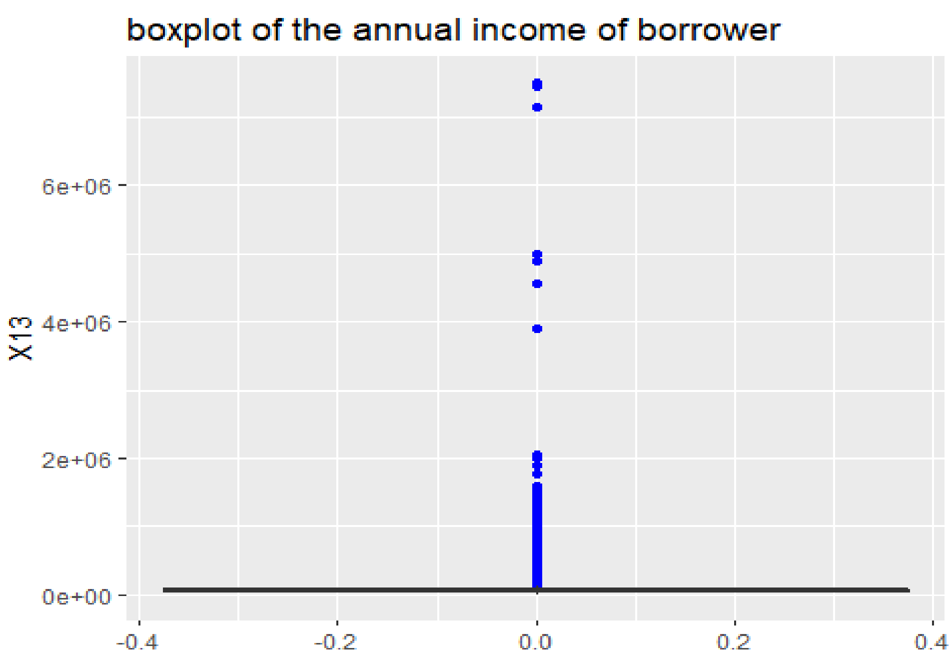

```
ggplot(load_data1, aes(x=X8))+geom_bar(stat = "count",aes(fill=X8))+facet_gri
d(~X12_cb)+labs( x="Loan Grade", y="Frequency")+ggtitle("distribution of the
grade among home ownership")
```



X13

X13, the annual income of the borrower, a numeric variable. Check the missing values and replace them with the mean.

```
sum(is.na(load_data1$X13))

## [1] 51751

summary(load_data1$X13)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    3000   45000   63000   73151   88079 7500000   51751

load_data1$X13[is.na(load_data1$X13)]=mean(load_data1$X13, na.rm = T)
sum(is.na(load_data1$X13))

## [1] 0

summary(load_data1$X13)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3000   48581   70000   73151   82000 7500000

ggplot(load_data1, aes(y=X13))+geom_boxplot(outlier.color = "blue")+ggtitle("
boxplot of the annual income of borrower")
```



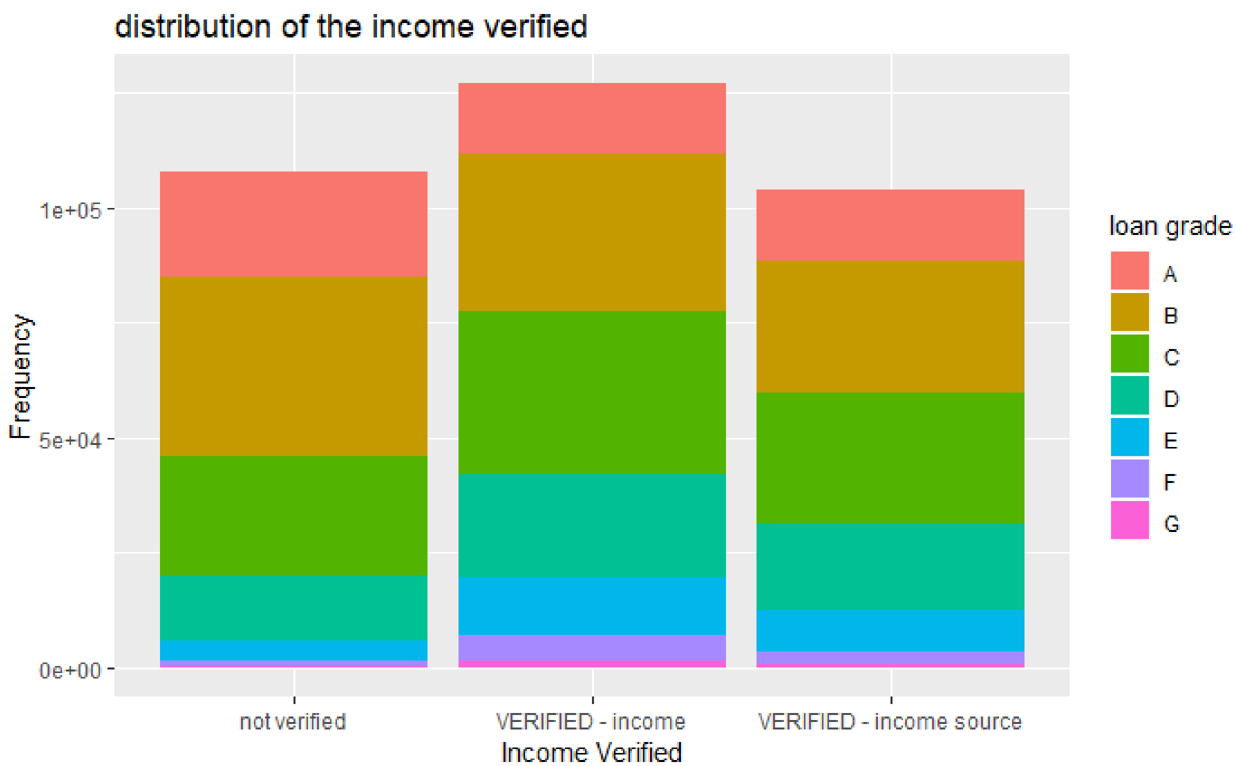
X14

X14, income source verified or not.

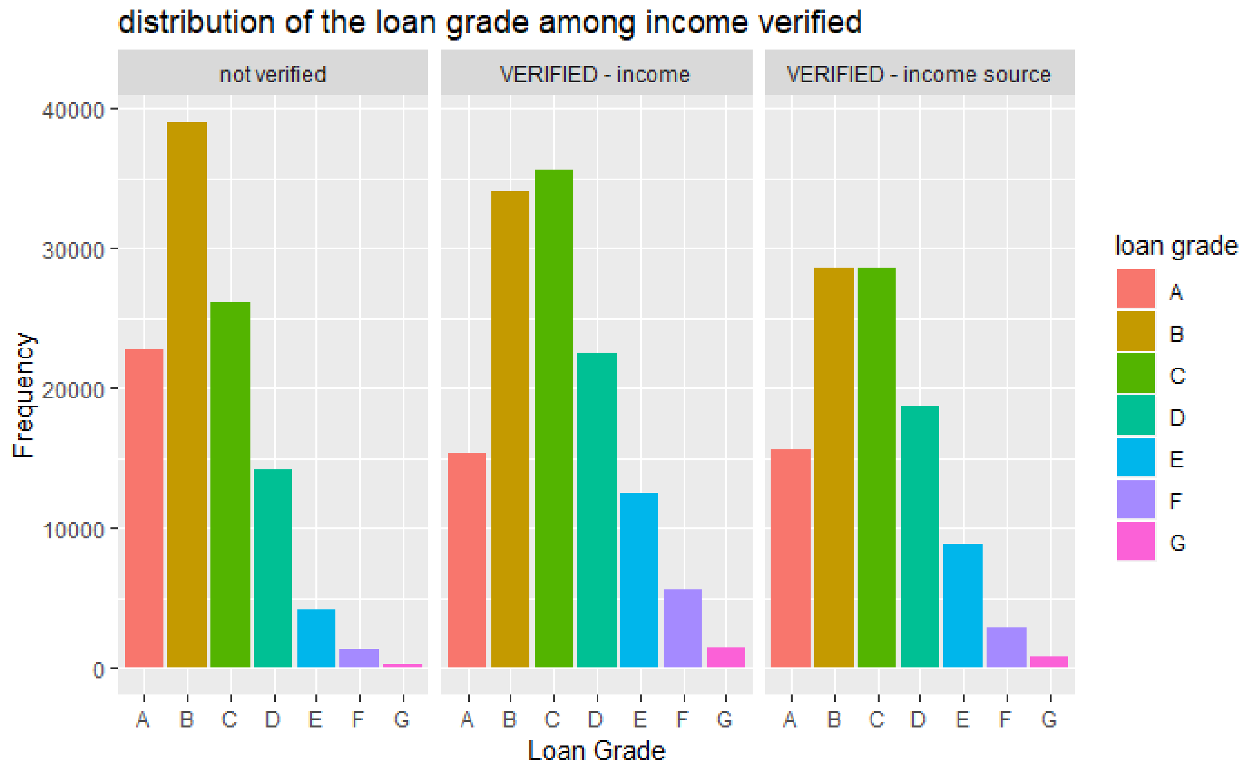
```
sum(is.na(load_data1$X14))
```

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

```r
load_data1$X14=droplevels(load_data1)$X14
x14_prop=prop.table(table(load_data1$X14))
ggplot(load_data1, aes(x=X14))+geom_bar(stat = 'count',aes(fill=X8))+labs(fil
l="loan grade", x="Income Verified", y="Frequency")+ggtitle("distribution of
the income verified")
```




```r
ggplot(load_data1, aes(x=X8))+geom_bar(stat = 'count',aes(fill=X8))+facet_gri
d(~X14)+labs(fill="loan grade", x="Loan Grade", y="Frequency")+ggtitle("distr
ibution of the loan grade among income verified")
```



x15

X15, date loan was issued. Time has an influence on market conditions. Extracting the year from the date values to focus on the impact of years.

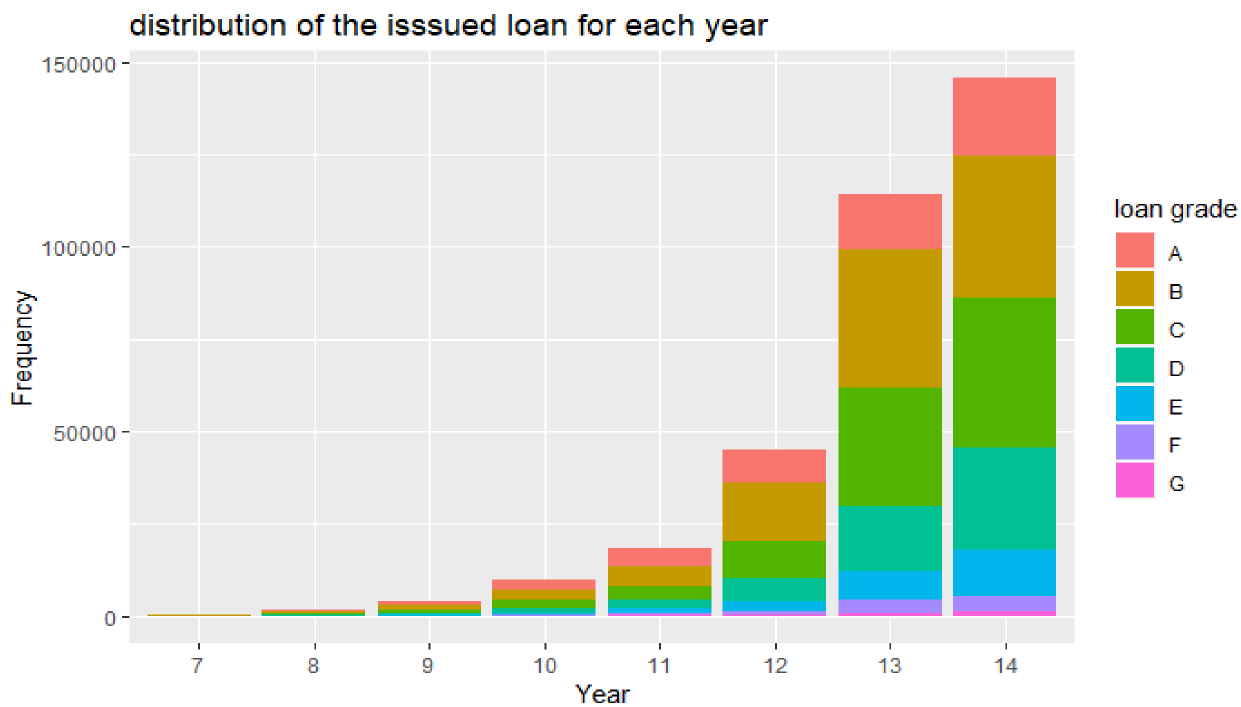
```
head(load_data1$X15, n=1)

## [1] 9-Aug
## 91 Levels: 10-Apr 10-Aug 10-Dec 10-Feb 10-Jan 10-Jul 10-Jun ... 9-Sep

library(stringr)
load_data1$X15_year=gsub("-*[A-Za-z]","",load_data1$X15)
head(load_data1$X15_year)

## [1] "9"  "8"  "14" "10" "9"  "12"

sum(is.na(load_data1$X15_year))

## [1] 0
```

The result below shows the number of issued loans increases every year. And every year the loan issued to borrowers with grade A comprises a sizeable proportion.

```
load_data1$X15_year=factor(droplevels(load_data1)$X15_year, levels = c("7","8
","9","10","11","12","13","14"))
table(load_data1$X15_year)
```

```
##
##      7      8      9     10     11     12     13     14
##    237   1517   4008   9792  18246  45289 114219 145681
```

```
ggplot(load_data1, aes(x=X15_year))+geom_bar(stat = 'count',aes(fill=X8))+lab
s(fill='loan grade',x="Year",y="Frequency")+ggtitle("distribution of the isss
ued loan for each year")
```



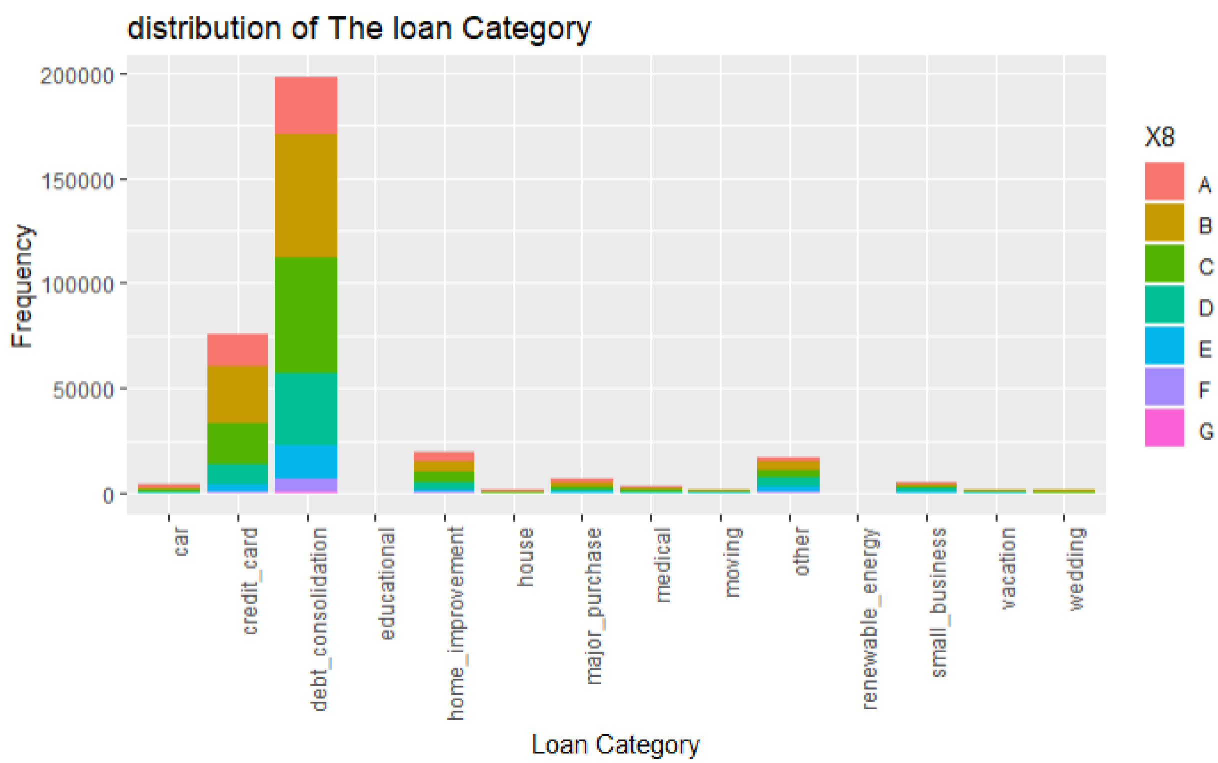
X16, X17 and X18

X16 (reasons for the loan), X17 (loan Category) and X18 (loan title) convey the same information. Hence, merely take X17 into consideration. Most people apply loan for debt consolidation.

```
#missing value of the loan categories
sum(is.na(load_data1$X17))

## [1] 0

# distribution of the loan categories
load_data1$X17=droplevels(load_data1)$X17
table(load_data1$X17)

##
##                car        credit_card debt_consolidation
##               4115              75680             198226
##        educational   home_improvement              house
##                279              19625               1723
##     major_purchase            medical             moving
```

```
##                7312                3329                2138
##               other    renewable_energy      small_business
##               17154                 267                5359
##            vacation             wedding
##                1848                1934
```

```
ggplot(load_data1, aes(x=X17))+geom_bar(stat = "count",aes(fill=X8))+labs(x="
Loan Category",y="Frequency", fill="loan category") + theme(axis.text.x = ele
ment_text(angle = 90, hjust = 1))+ggtitle("distribution of The loan Category"
)
```



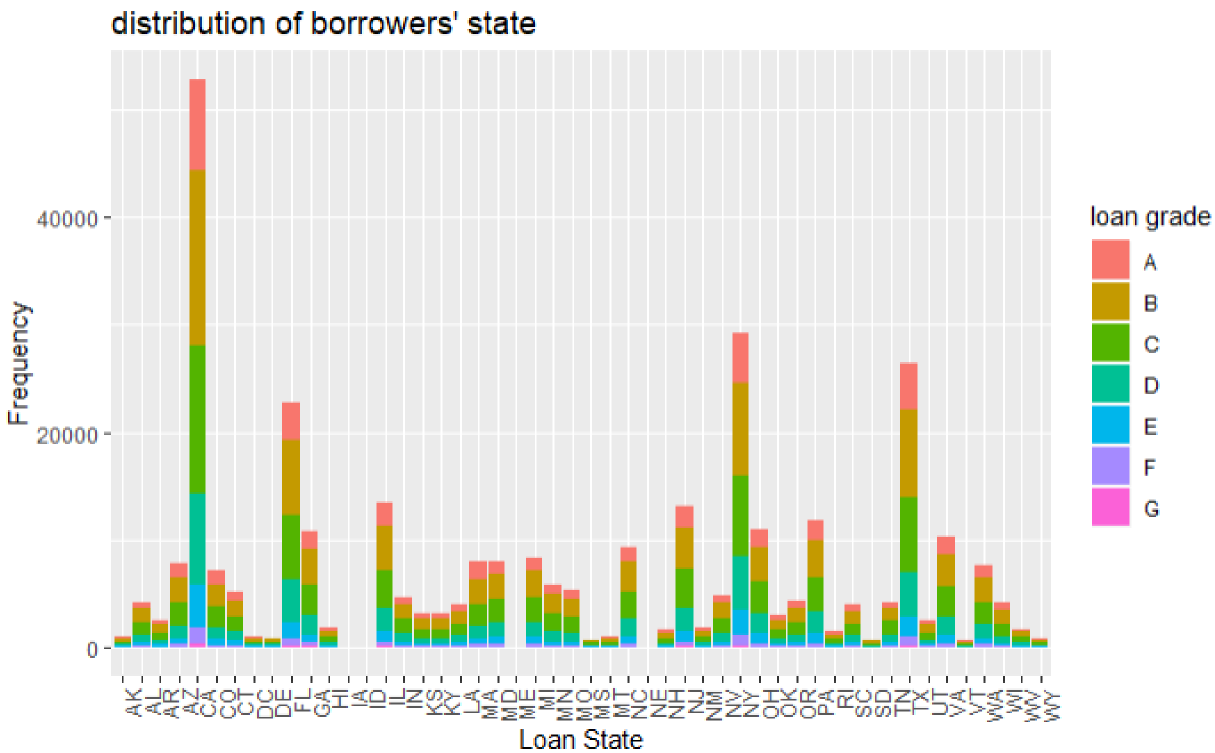
X19

X19 is the state of the borrower, from the distribution of the state of borrowers. People have high frequencies of requesting the loan in CA, NY, TX, FL

```
sum(is.na(load_data1$X19))
```

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

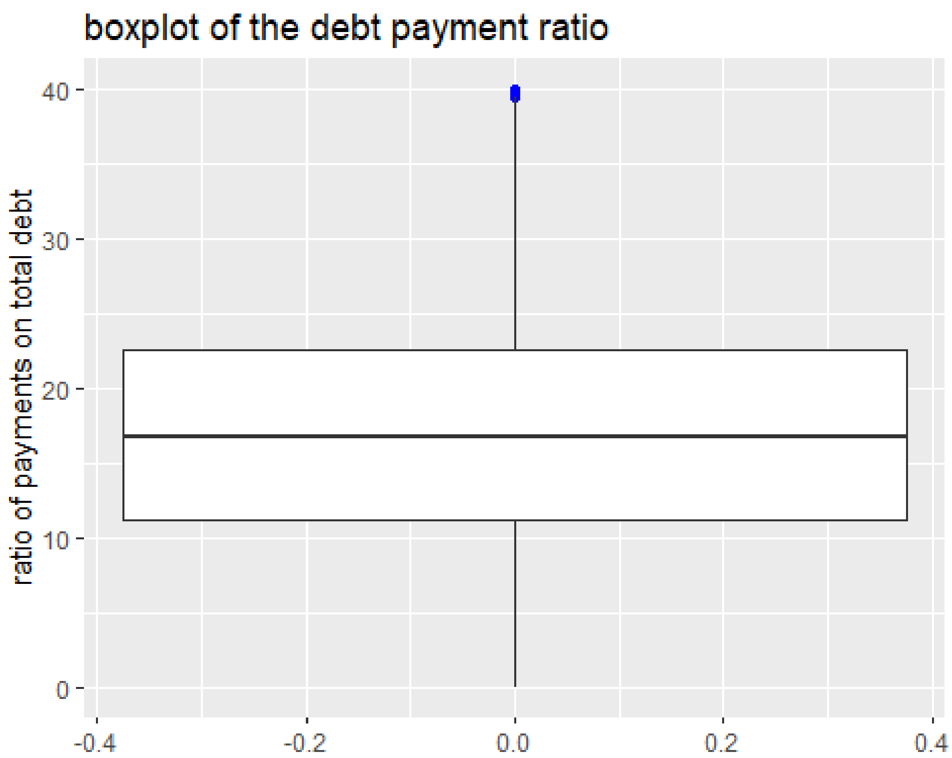
```
load_data1$X19=droplevels(load_data1)$X19
ggplot(load_data1, aes(x=X19))+geom_bar(stat = "count",aes(fill=X8))+labs(fil
l='loan grade',x="Loan Category",y="Frequency") + theme(axis.text.x = element
_text(angle = 90, hjust = 1))+ggtitle("distribution of borrowers' state")
```



X20

X20, the ratio calculated employing the borrower's total monthly debt payments on the total debt obligations, is a numeric factor. The minimal payment ratio is 0, and the maximal payment ratio is 39.99. Mean of payment ratio is 17.00.

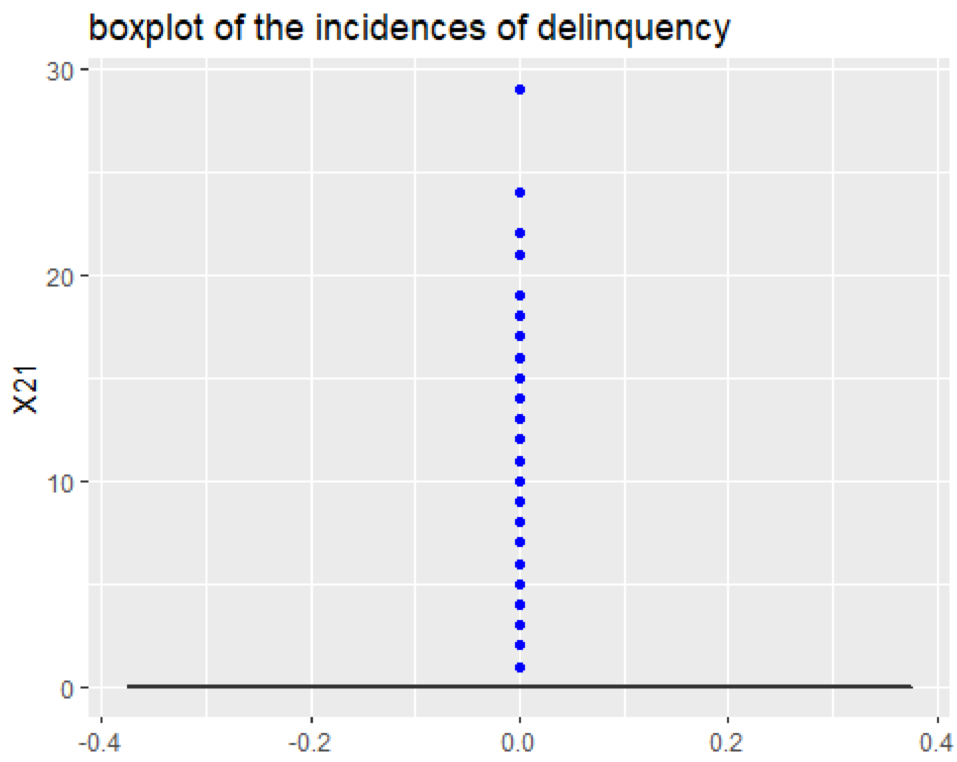
```
sum(is.na(load_data1$X20))

## [1] 0

summary(load_data1$X20)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   11.25   16.70   17.00   22.50   39.99

ggplot(load_data1, aes(y=X20))+geom_boxplot(outlier.color = "blue")+ggtitle("
boxplot of the debt payment ratio")+labs(y="ratio of payments on total debt")
```



X21

X21, the number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years, is a numeric variable as well. The minimal number of incidences is 0; the maximum is 29. Mean is 0.2745. However, most borrowers have 0 delinquencies.

```
sum(is.na(load_data1$X21))

## [1] 0

load_data1$X21=as.numeric(load_data1$X21)
summary(load_data1$X21)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.0000  0.0000  0.2743  0.0000 29.0000

ggplot(load_data1, aes(y=X21))+geom_boxplot(outlier.color = "blue")+ggtitle("
boxplot of the incidences of delinquency")
```



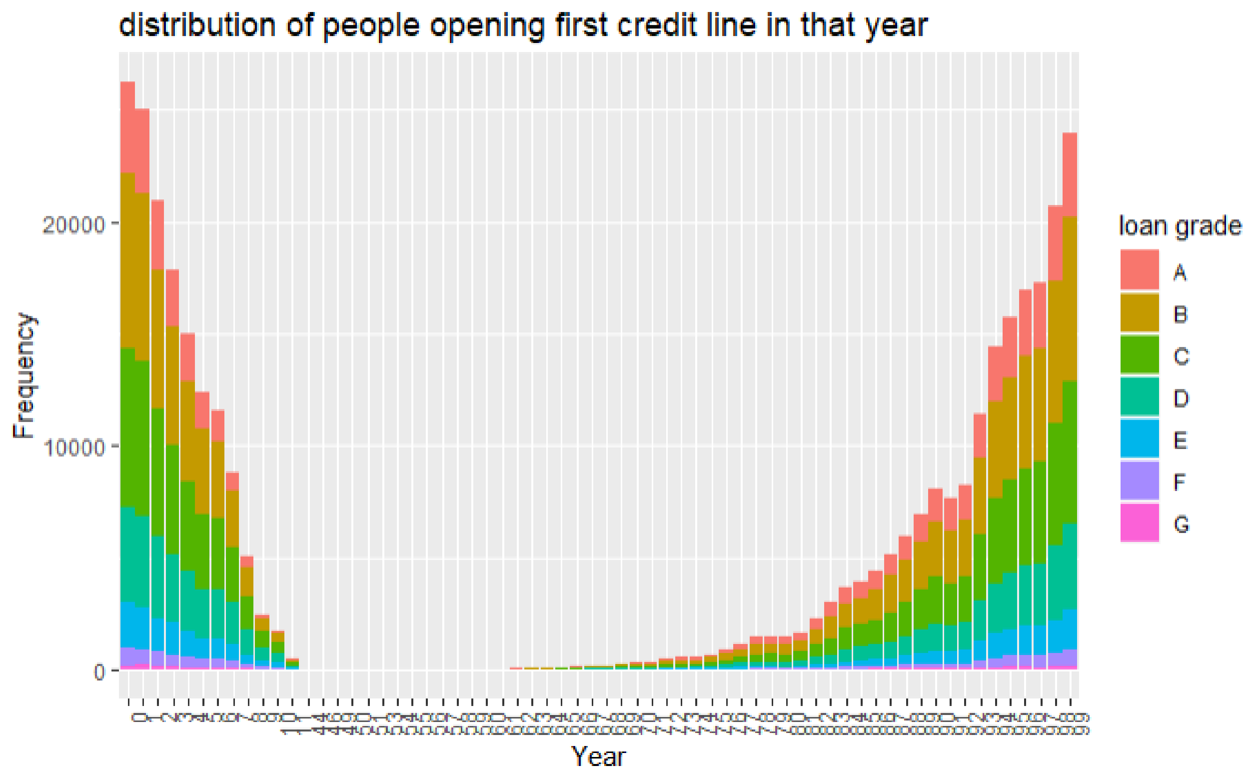
X22

X22, the date the borrower's earliest reported credit line was opened, is a categorical variable with date values. Same as X15, extract the year from these date values.

```
sum(is.na(load_data1$X22))

## [1] 0

library(stringr)
numextract <- function(string){
  str_extract(string, "\\-*\\d+\\.*\\d*")

}
load_data1$X22_year=as.numeric(gsub("-","",numextract(load_data1$X22)))

head(load_data1$X22_year, n=20)

##  [1] 94  0  0 85 96 94  0 98 93  1  6 95 97  1 90  0 96 91 98  3

sum(is.na(load_data1$X22_year))

## [1] 0
```

From the distribution, the result shows that the number of borrowers opening the first credit line increase from 1944 to 2000 and decrease from 2000 to 2011.

```
load_data1$X22_year=factor(droplevels(load_data1)$X22_year)
# distribution of people opening the first credit line in that year
ggplot(load_data1, aes(x=X22_year, fill=X8))+geom_bar(stat = 'count')+labs(fi
ll="loan grade", x="Year",y="Frequency")+theme(axis.text.x = element_text(ang
le = 90, hjust = 1))+ggtitle("distribution of people opening first credit lin
e in that year")
```



X23, X24

X23, number of months since the borrower's last delinquency, is a numeric variable. The portion of missing value arrives at 55.4%. In that case, unambiguously, drop this variable. X24, number of months since the last public record, is withal a numeric variable. The portion of missing value arrives at 87.2%, so drop this variable as well.

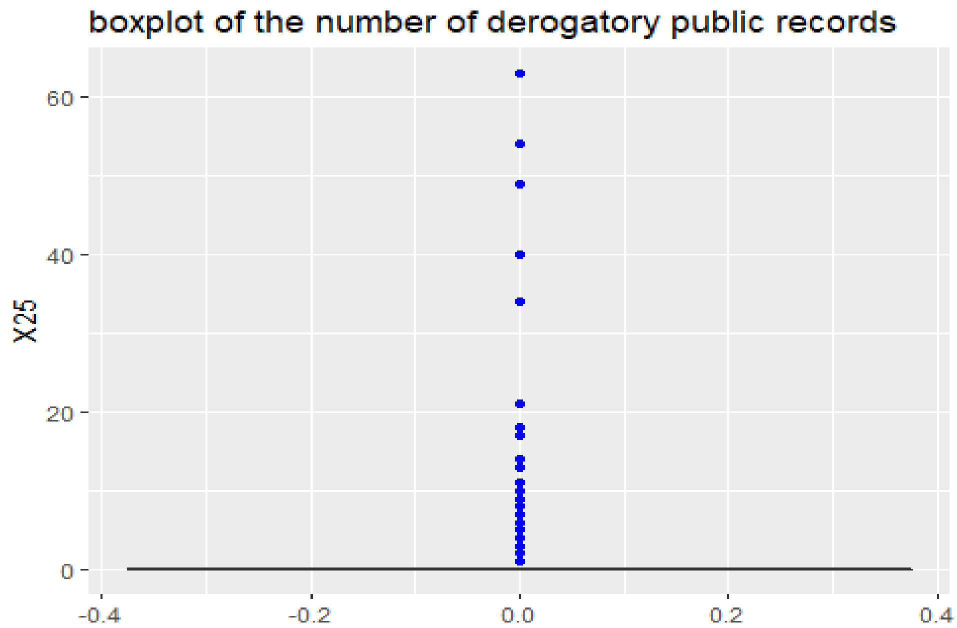
```
sum(is.na(load_data1$X23))

## [1] 185456

summary(load_data1$X23)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   16.00   31.00   34.33   50.00  188.00  185456

sum(is.na(load_data1$X23))/sum(!is.na(load_data1$X22))

## [1] 0.5470856
```

```
sum(is.na(load_data1$X24))

## [1] 295589

summary(load_data1$X24)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   54.00   80.00   76.16  103.00  129.00  295589

sum(is.na(load_data1$X24))/sum(!is.na(load_data1$X22))

## [1] 0.8719722
```

X25

X25, the number of derogatory public records. The minimal number of derogatory is 0, and the maximum is 63. Mean is 0.1532. However, based on the boxplot, only 50485 observations have the number of derogatory public records non-zero.

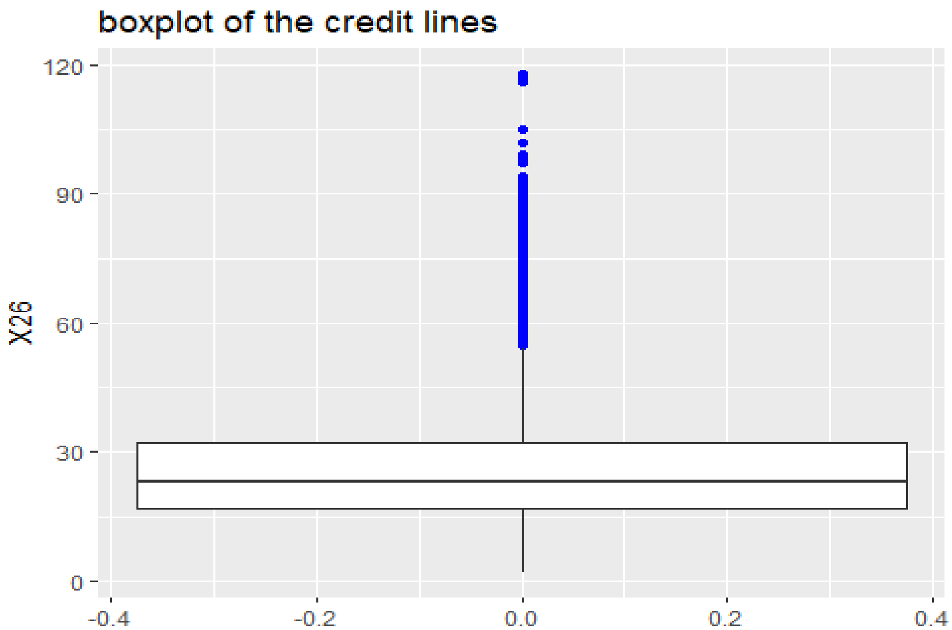
```
sum(is.na(load_data1$X25))

## [1] 0

load_data1$X25=as.numeric(load_data1$X25)
sum(load_data1$X25!=0)

## [1] 42760

summary(load_data1$X25)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.1527  0.0000 63.0000

ggplot(load_data1, aes(y=X25))+geom_boxplot(outlier.color = "blue")+ggtitle("
boxplot of the number of derogatory public records")
```



X26

X26, the total number of credit lines currently in the borrower's credit file. The minimum is 2; mean is 25; the maximum is 118.

```
sum(is.na(load_data1$X26))

## [1] 0

load_data1$X26=as.numeric(load_data1$X26)
summary(load_data1$X26)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    2.00   17.00   23.00   24.98   32.00  118.00

ggplot(load_data1, aes(y=X26))+geom_boxplot(outlier.color = "blue")+ggtitle("
boxplot of the credit lines")
```

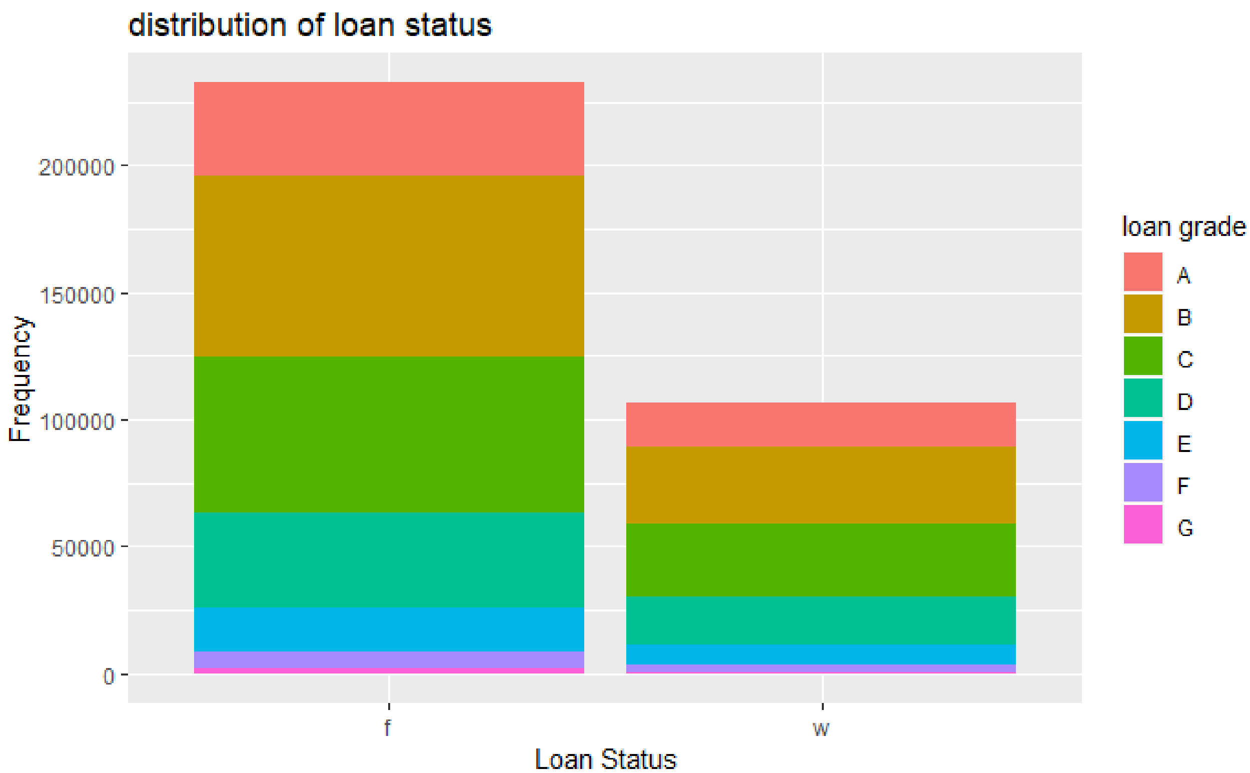



```
# take a look at the person with 118 credit lines
load_data1[load_data1$X26==118,]

##            X1   X4   X5   X6         X7 X8 X9                     X10
## 364271 13.98 5000 5000 5000  36 months  C C3 Mental Health Clinician
##           X11  X12   X13           X14    X15  X16                X17
## 364271 1 year <NA> 90000 not verified 14-Aug <NA> debt_consolidation
##                      X18 X19   X20 X21    X22 X23 X24 X25 X26 X27 X12_cb
## 364271 Debt consolidation  CA 27.63   0 Jul-98  NA  NA   0 118   w  OTHER
##        X15_year X22_year
## 364271       14       98
```

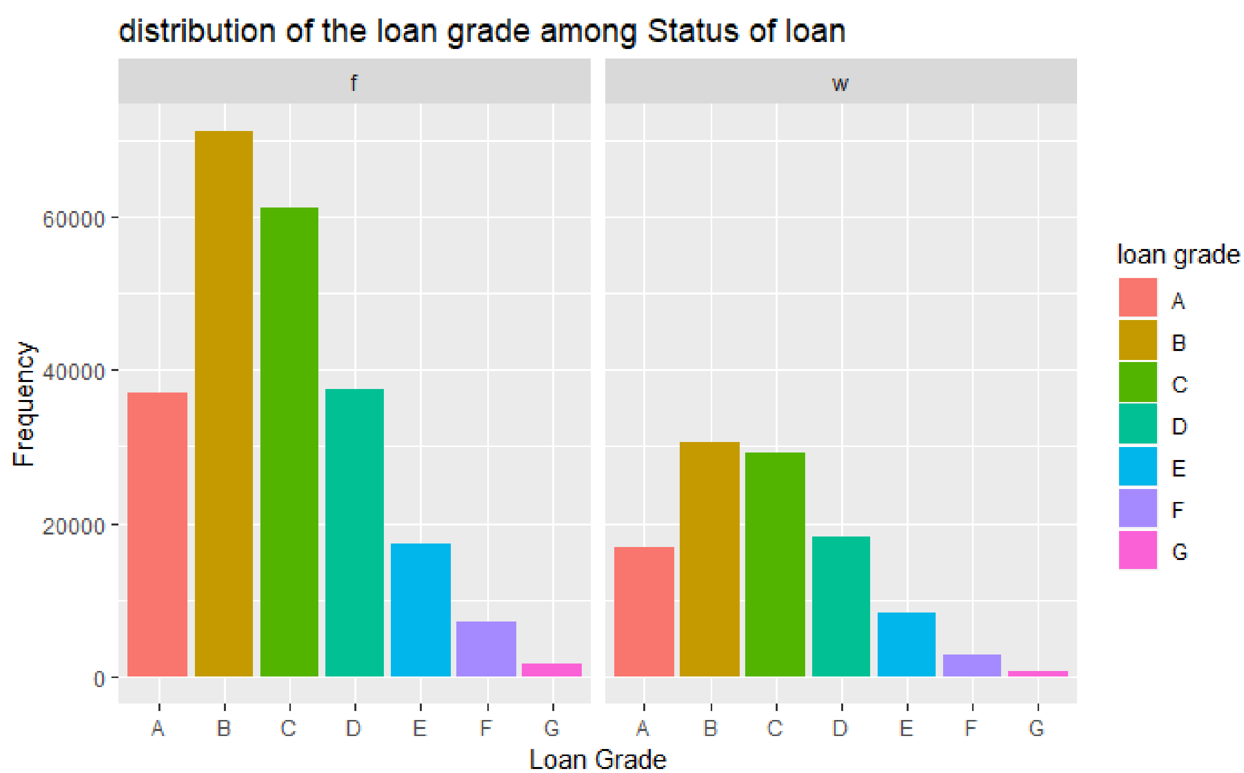
X27

X27, the initial listing status of the loan, includes two levels: "W" and "F".

```
sum(is.na(load_data1$X27))

## [1] 0

load_data1$X27=droplevels(load_data1)$X27
ggplot(load_data1, aes(x=X27, fill=X8))+geom_bar(stat = "count")+labs(fill="l
oan grade" ,x="Loan Status",y="Frequency")+ggtitle("distribution of loan stat
us")
```




```
ggplot(load_data1, aes(x=X8, fill=X8))+geom_bar(stat = 'count')+facet_grid(~X
27)+labs( x="Loan Grade",fill="loan grade", y="Frequency")+ggtitle("distribut
ion of the loan grade among Status of loan")
```



1.2 Reform data frame and normalize numeric variables

There are several numeric variables with diverse scales. To decrease the scaling influence for prediction, apply the z-score standardization as some variables have extreme outliers.

```
dim(load_data1)

## [1] 338989     28

load_data2=within(load_data1, rm(X10,X12,X15,X16,X18,X22,X23,X24))
str(load_data2)

## 'data.frame':    338989 obs. of  20 variables:
##  $ X1      : num  11.9 10.7 17 13.1 13.6 ...
##  $ X4      : num  25000 7000 25000 1200 10800 7200 7500 3000 4000 5600 ...
##  $ X5      : num  25000 7000 25000 1200 10800 ...
##  $ X6      : num  19080 673 24725 1200 10692 ...
##  $ X7      : Factor w/ 2 levels " 36 months"," 60 months": 1 1 1 1 1 1 1 1
 1 1 ...
##  $ X8      : Factor w/ 7 levels "A","B","C","D",..: 2 2 4 3 3 4 2 3 1 4 ..
.
##  $ X9      : Factor w/ 35 levels "A1","A2","A3",..: 9 10 18 12 13 19 8 15
5 17 ...
##  $ X11     : Factor w/ 12 levels "< 1 year","1 year",..: 1 1 2 3 8 11 5 5
1 2 ...
##  $ X13     : num  85000 65000 70000 54000 32000 58000 85000 80800 148000 4
5000 ...
##  $ X14     : Factor w/ 3 levels "not verified",..: 2 1 2 1 1 3 1 1 1 1 ...
##  $ X17     : Factor w/ 14 levels "car","credit_card",..: 3 2 3 3 3 3 3 2 2
 3 ...
##  $ X19     : Factor w/ 50 levels "AK","AL","AR",..: 5 34 34 43 7 39 5 43 4
3 21 ...
##  $ X20     : num  19.48 14.29 10.5 5.47 11.63 ...
##  $ X21     : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ X25     : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ X26     : num  42 7 17 31 40 25 11 23 19 9 ...
##  $ X27     : Factor w/ 2 levels "f","w": 1 1 1 1 1 1 1 1 1 1 ...
##  $ X12_cb  : Factor w/ 4 levels "MORTGAGE","OTHER",..: 4 4 4 3 4 4 4 1 1 4
 ...
##  $ X15_year: Factor w/ 8 levels "7","8","9","10",..: 3 2 8 4 3 6 2 3 4 4 .
..
##  $ X22_year: Factor w/ 64 levels "0","1","2","3",..: 59 1 1 50 61 59 1 63
58 2 ...

dim(load_data2)

## [1] 338989     20

num_df=c('X4', 'X5', 'X6','X13','X20','X21','X25','X26')
cat_var=c('X7','X8','X9','X11','X14','X17','X19','X27','X12_cb','X15_year','X
22_year')
```

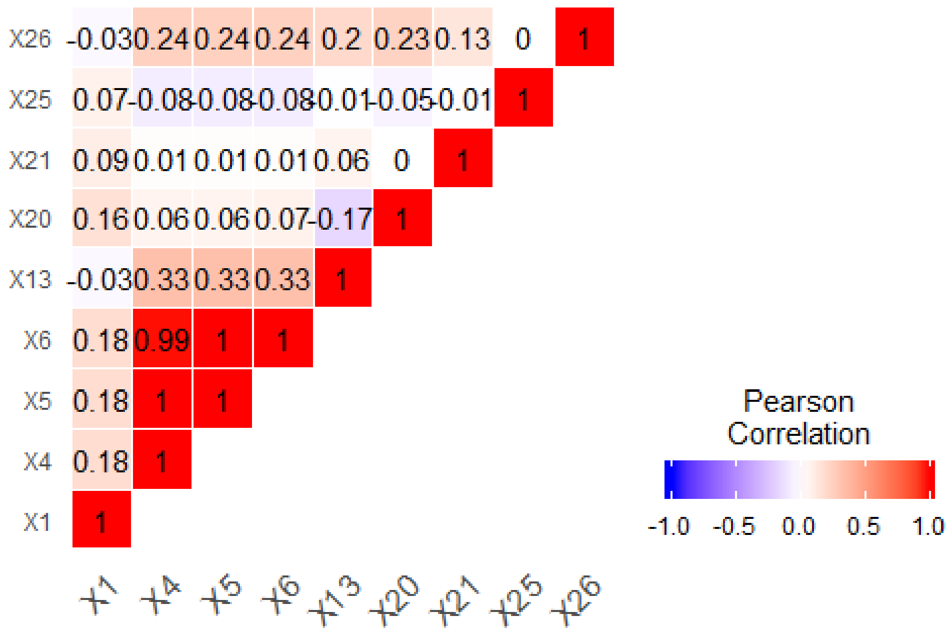
```
load_data3_num=as.data.frame( scale(load_data2[num_df] ))
load_data3=cbind(load_data2$X1,load_data3_num, load_data2[,cat_var] )
names(load_data3)[names(load_data3)=="load_data2$X1"]="X1"
```

Now, there is a dataset with normalized numeric variables and categorical variables named "load_data3".

2. Testing the significance of variables

2.1 Correlation among numeric variables

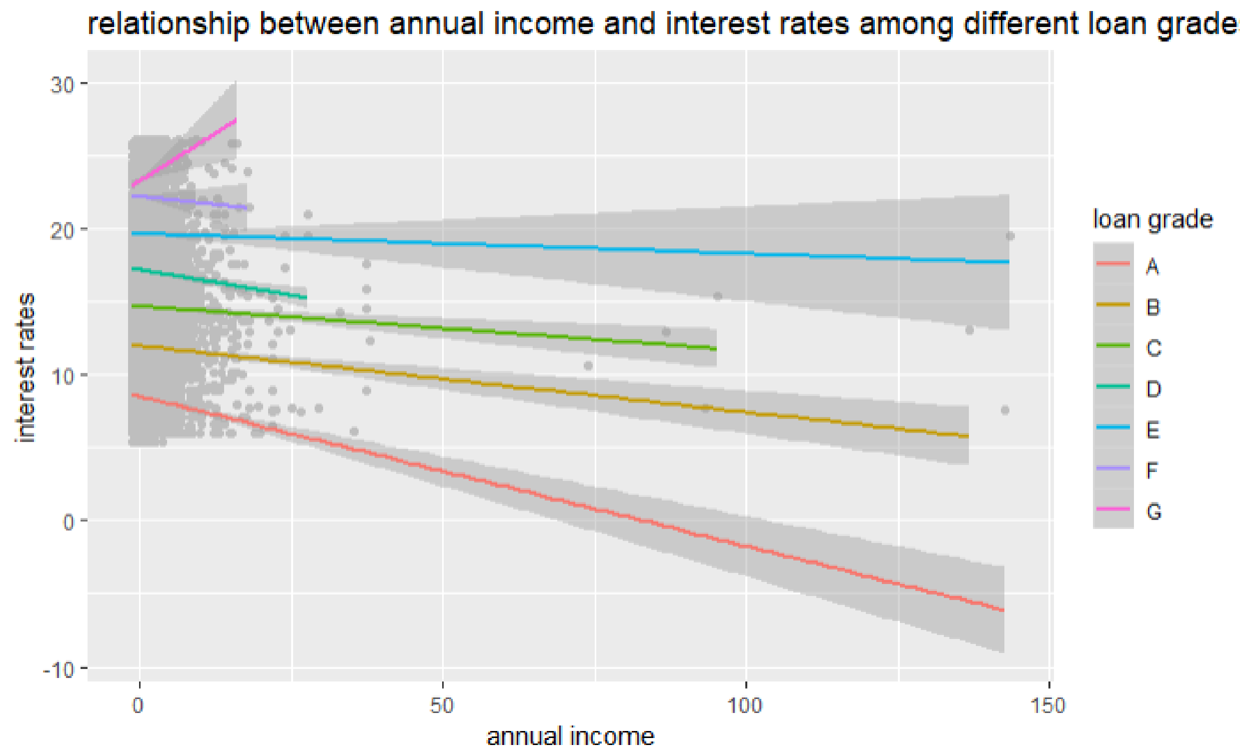
```
cor=round(cor(load_data3[,c("X1",num_df)]),2)
library(reshape2)

cor[upper.tri(cor)]=NA
cormat=melt(cor, na.rm = T)
library(ggplot2)
##correlation matrix heatmap
ggplot(data = cormat, aes(Var2, Var1, fill = value))+
 geom_tile(color = "white")+
 scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
   name="Pearson\nCorrelation") +
  theme_minimal()+
 theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 12, hjust = 1))+
 coord_fixed()+geom_text(aes(Var2, Var1, label = value), color = "black", siz
e = 4) +
theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  panel.grid.major = element_blank(),
  panel.border = element_blank(),
  panel.background = element_blank(),
  axis.ticks = element_blank(),
  legend.justification = c(1, 0),
  legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                title.position = "top", title.hjust = 0.5))
```



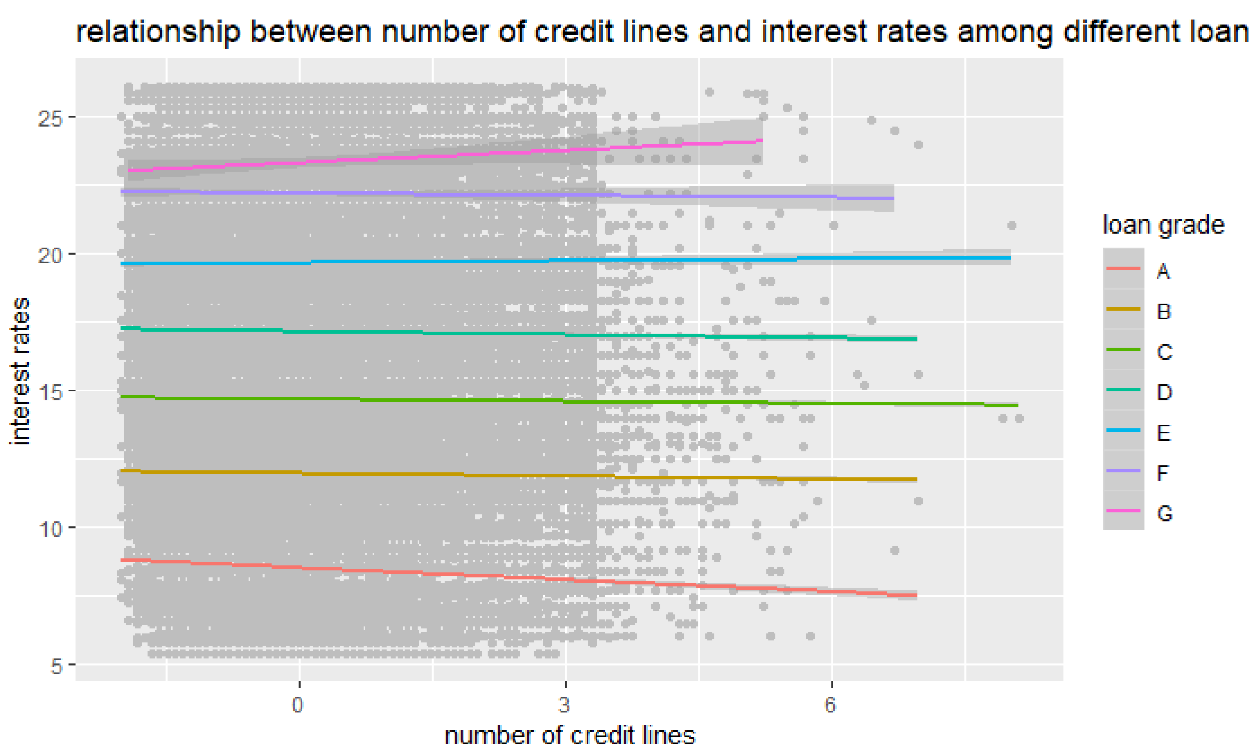
From the previous result, correlation coefficients of X4, X5, and X6 equal to 1, which means these variables are collinear. X1 and X13 demonstrate a negative correlation, which means borrowers maintain high annual income will get a lower interest rate. It makes sense people with more upper income prove more able to pay off loans. X1 and X26 as well exhibit a negative correlation. The more credit lines borrowers open, the lower interest rates they will get when applying for the loans. It is possible people with a great number of credit lines will have a good loan grade.

Let's perceive the relationship among the loan grade, annual income, and interest rates, as well as loan grade, the numbers of credit lines and interest rates. From the relationships, when borrowers have level A loan grade, the higher annual income they gain, the lower interest rates they get. Also, the more credit lines they open, the interest rates are lower. However, for borrowers from the level G group, people with higher income or more credit lines have higher interest rates.

```
# graphs of relationship
ggplot(data=load_data3, aes(x=X13, y=X1, color=X8))+geom_point(color="grey")+
geom_smooth(method="lm")+labs(x="annual income", y="interest rates", color="l
oan grade")+ggtitle("relationship between annual income and interest rates am
ong different loan grades")
```




```r
ggplot(data=load_data3, aes(x=X26, y=X1, color=X8))+geom_point(color="grey")+
geom_smooth(method="lm")+labs(x="number of credit lines", y="interest rates",
 color="loan grade")+ggtitle("relationship between number of credit lines and
 interest rates among different grades")
```

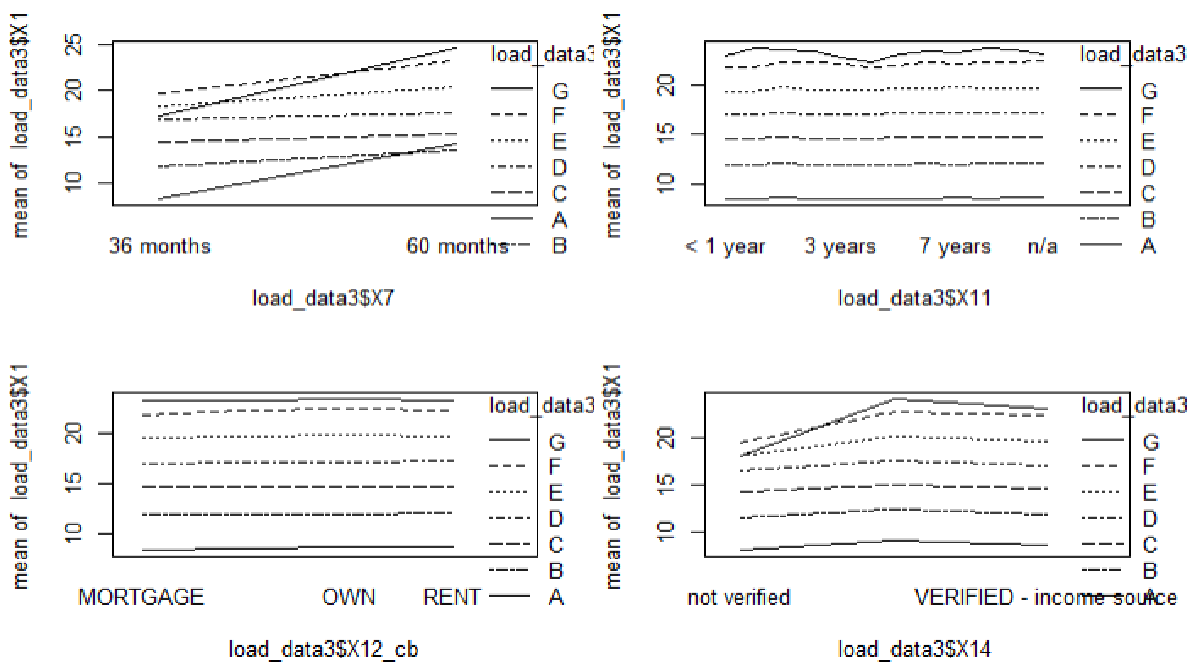


2.2 Significance of the numerical variables

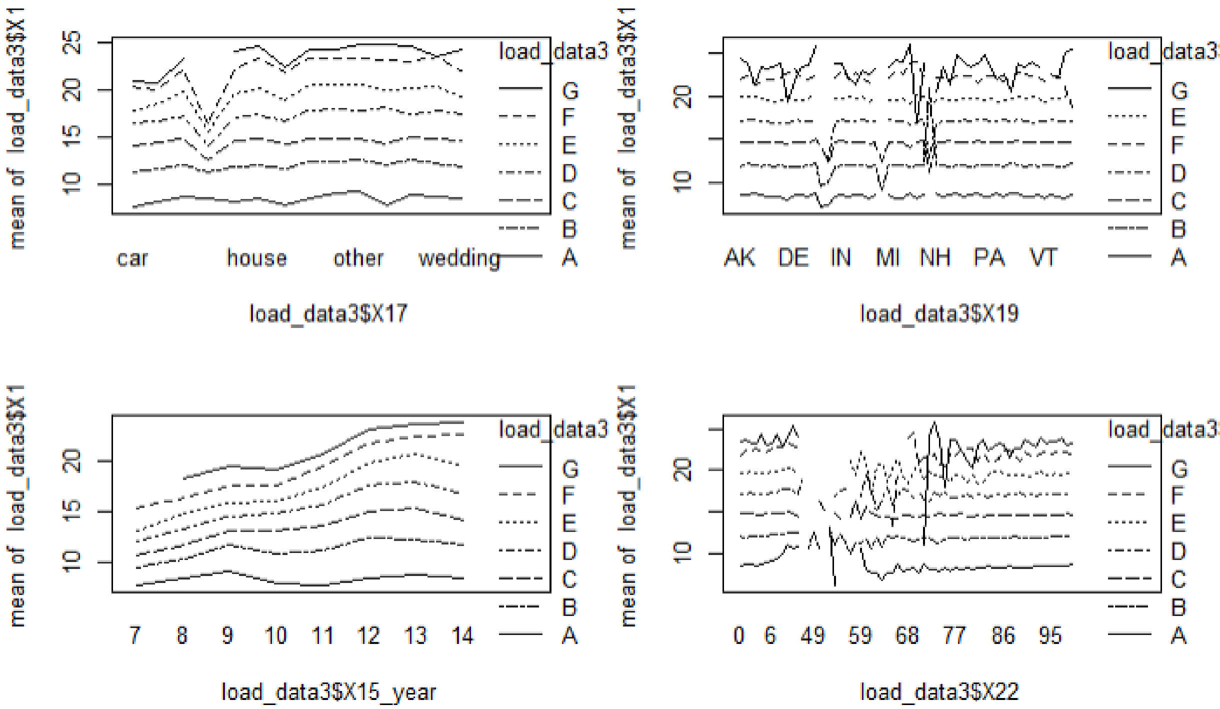
```
fm=aov(X1~X4+X13+X20+X21+X25+X26, data=load_data3)
summary(fm)

##                 Df  Sum Sq Mean Sq F value Pr(>F)    
## X4               1  205238  205238   11744 <2e-16 ***
## X13              1   56750   56750    3247 <2e-16 ***
## X20              1  112431  112431    6434 <2e-16 ***
## X21              1   57962   57962    3317 <2e-16 ***
## X25              1   58434   58434    3344 <2e-16 ***
## X26              1   82400   82400    4715 <2e-16 ***
## Residuals   338982 5923965      17                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.3 Interaction effect, and interests among different groups

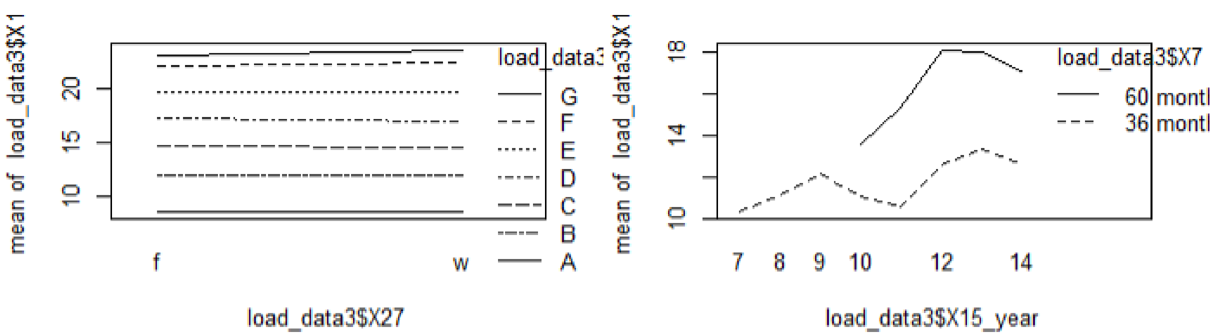
```
# interaction among varibles and loan grade
par(mfrow=c(2,2))
interaction.plot(load_data3$X7, load_data3$X8, load_data3$X1)
interaction.plot(load_data3$X11, load_data3$X8, load_data3$X1)
interaction.plot(load_data3$X12_cb, load_data3$X8, load_data3$X1)
interaction.plot(load_data3$X14, load_data3$X8, load_data3$X1)
```




```
interaction.plot(load_data3$X17, load_data3$X8, load_data3$X1)
interaction.plot(load_data3$X19, load_data3$X8, load_data3$X1)
interaction.plot(load_data3$X15_year, load_data3$X8, load_data3$X1)
interaction.plot(load_data3$X22, load_data3$X8, load_data3$X1)
```




```
interaction.plot(load_data3$X27, load_data3$X8, load_data3$X1)
interaction.plot(load_data3$X15_year, load_data3$X7, load_data3$X1)
par(mfrow=c(1,1))
```



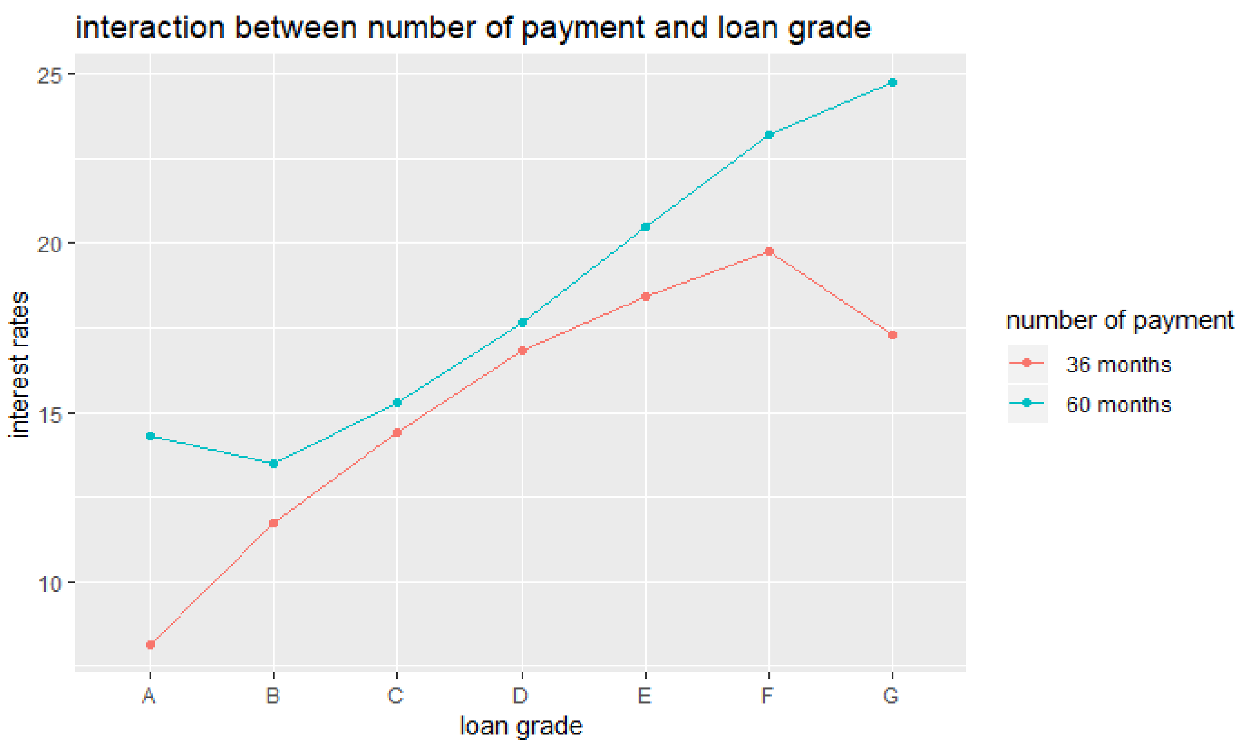
Within the same group, borrowers who have 60 months payments tend to have higher interest rates, and when borrowers have the same number of payments, greater level loan grades they have, lower interest rates they will be requested.

```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode
```

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

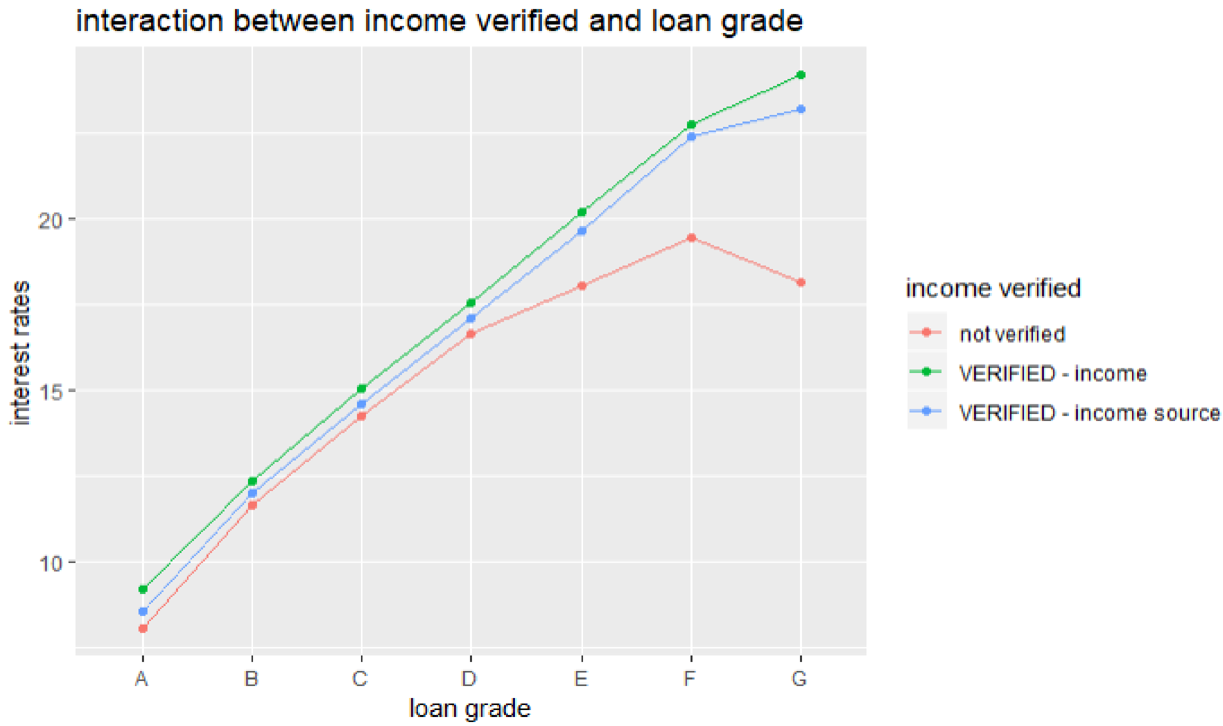
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# interaction between the number of payment and loan grade and their effects
on interest rates
tmp=load_data3 %>% group_by(X8,X7) %>% summarise(interest=mean(X1))
ggplot(data=tmp, aes(x=X8, y=interest, color=X7))+geom_line(aes(group=X7))+ge
om_point()+labs(x="loan grade", y="interest rates", color="number of payment"
)+ggtitle("interaction between number of payment and loan grade")
```



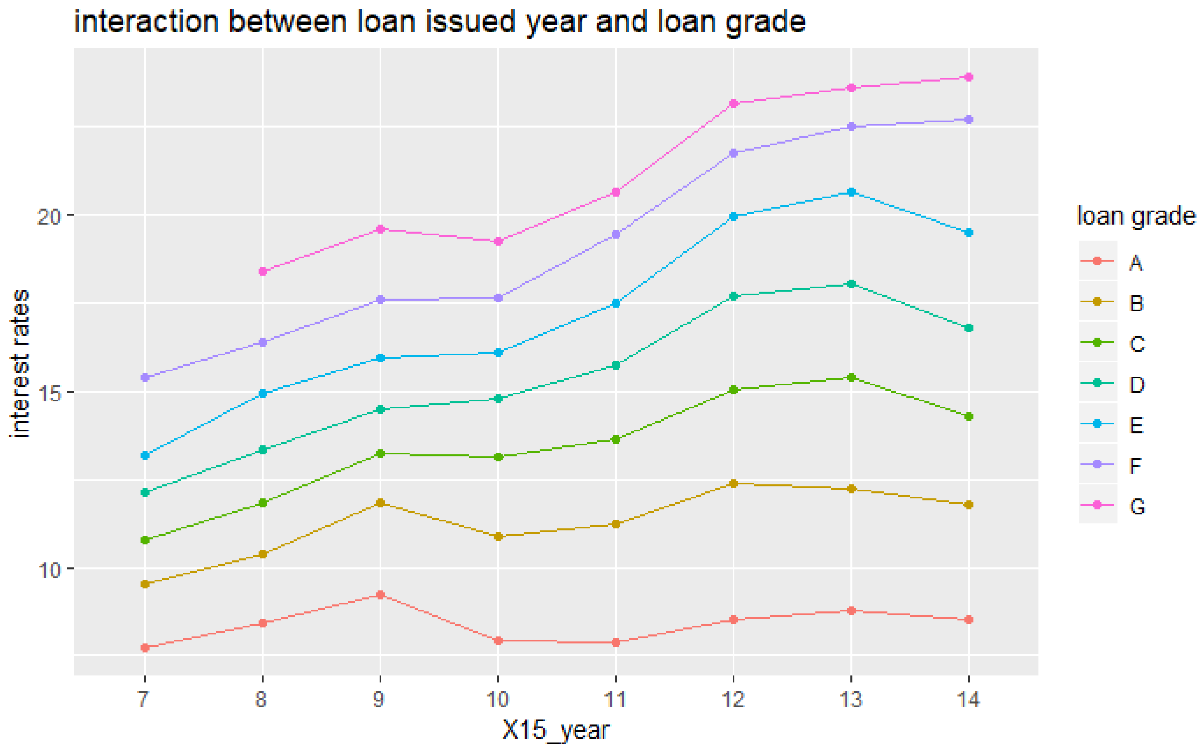
Within the same loan grade group, borrowers with verified income or income source have almost the same interest rate. Borrowers whose incomes are unverified tend to have a lower interest rate. But the difference is not too much.

```
## interaction between income verified and loan grade
tmp=load_data3 %>% group_by(X8,X14) %>% summarise(interest=mean(X1))
ggplot(data=tmp, aes(x=X8, y=interest, color=X14))+geom_line(aes(group=X14))+
geom_point()+labs(color="income verified", y="interest rates", x="loan grade"
)+ggtitle("interaction between income verified and loan grade")
```



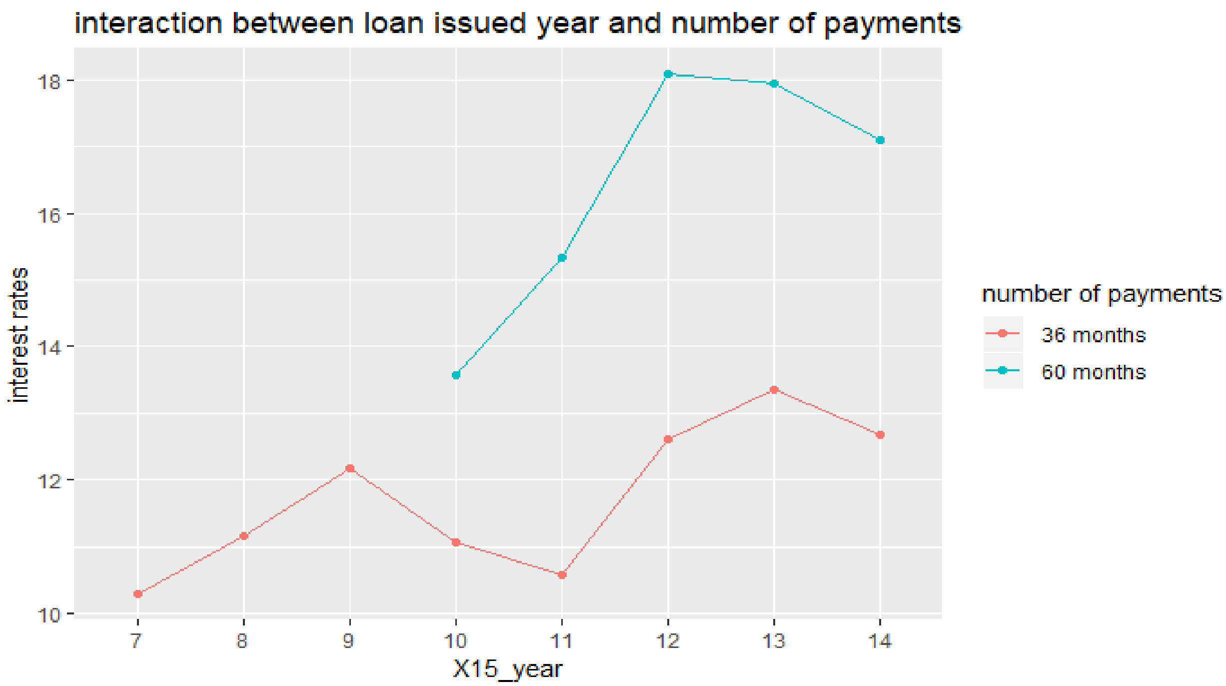
Within the same loan issued year, obviously that borrowers have better loan grade will get lower interest rates. And interest rates for the same loan grade fluctuate every year. Interest rates went up from 2008 to 2009 and fell from 2009 to 2010, and then went up again from 2010 to 2014.

```
tmp=load_data3 %>% group_by(X8,X15_year) %>% summarise(interest=mean(X1))
ggplot(data=tmp, aes(x=X15_year, y=interest, color=X8))+geom_line(aes(group=X
8))+geom_point()+labs(X=" loan issued year", y="interest rates", color="loan
grade")+ggtitle("interaction between loan issued year and loan grade")
```



The interest rates for borrowers in 60 months of payments group fluctuate every year. And the trends of fluctuations went up from 2007 to 2009 and fell from 2009 to 2010. Then went up again. Before 2010, there were no records for borrowers in 60 months of payment group. Interest rates for 60 months of payments went up from 2010 to 2012 and then fell to 2014.

```
# interaction between loan issued year and number of payments
tmp=load_data3 %>% group_by(X7,X15_year) %>% summarise(interest=mean(X1))
ggplot(data=tmp, aes(x=X15_year, y=interest, color=X7))+geom_line(aes(group=X
7))+geom_point()+labs(X=" loan issued year", y="interest rates", color="numbe
r of payments")+ggtitle("interaction between loan issued year and number of p
ayments")
```



From the previous results, clearly see that, within the same grade, level interest rates for diverse groups of work experience, home ownership status, loan category, state of the borrower, the date of the borrower's earliest reported credit line was opened virtually keep the same and loan status. Variables that exert apparent effects are a number of payments, income verified or not, and loan issued years.

So far, the relatively important variables are X4, X8/X9, X13, X20, X21, X25, X26, X7, X14, X15_year.

3. Sampling

3.1 Sampling by group

Sampling based on loan groups. The supposed sample size is 5000, according to the distribution of the loan grade, the number of sample extract from each group can be counted.

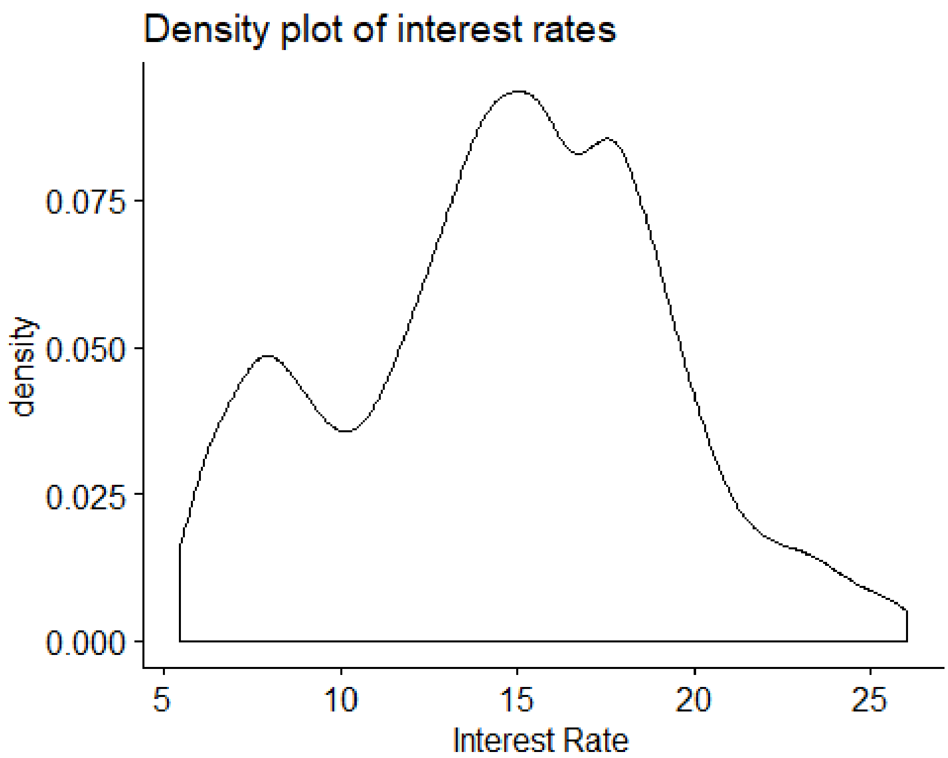
```
library(dplyr)
library(purrr)

##
## Attaching package: 'purrr'

## The following object is masked from 'package:car':
##
##     some
```

```
library(tidyr)

##
## Attaching package: 'tidyr'

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

set.seed(9933)
5000*round(x8_prop,3)

##
##    A    B    C    D    E    F    G
##  795 1500 1330  820  375  145   35

sig=c('X1', 'X8','X4','X13', 'X20', 'X21', 'X25', 'X26', 'X7', 'X14', 'X15_ye
ar')
sample_data=load_data3[,sig]%>% group_by(X8) %>% nest() %>% mutate(n=c(795,15
00,1330,820,375,145,35)) %>% mutate(samp=map2(data, n, sample_n)) %>% select(
X8, samp) %>% unnest()
dim(sample_data)

## [1] 5000   11
```

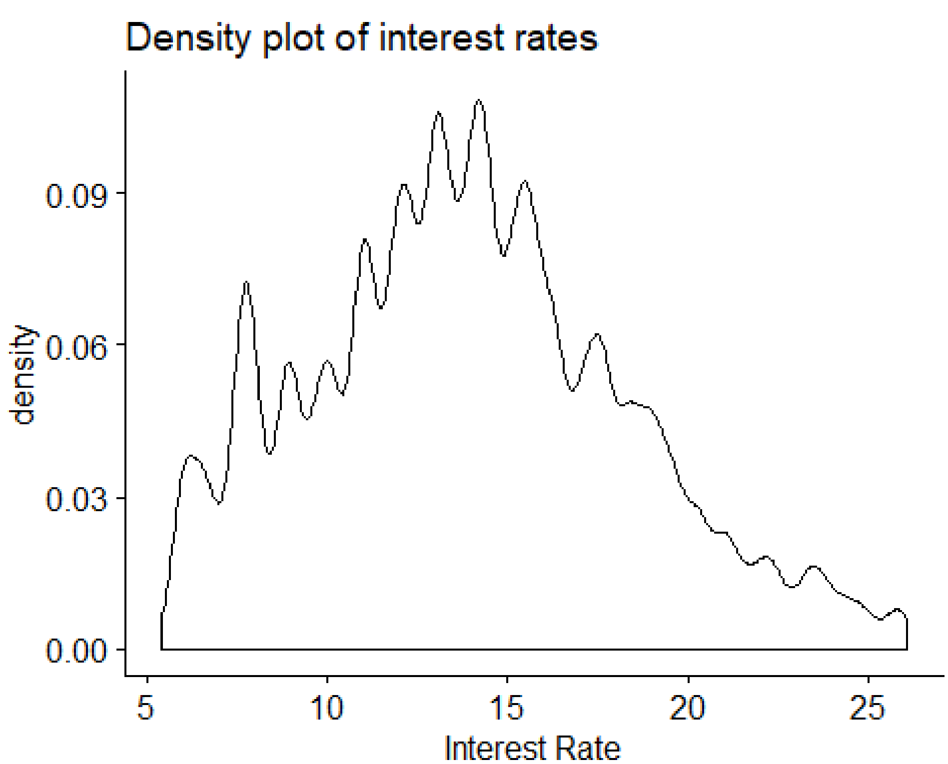
3.2 Inference of sample

The result shows the distribution of the X1 from the sample is almost the same as the distribution of X1 from the load_data3.

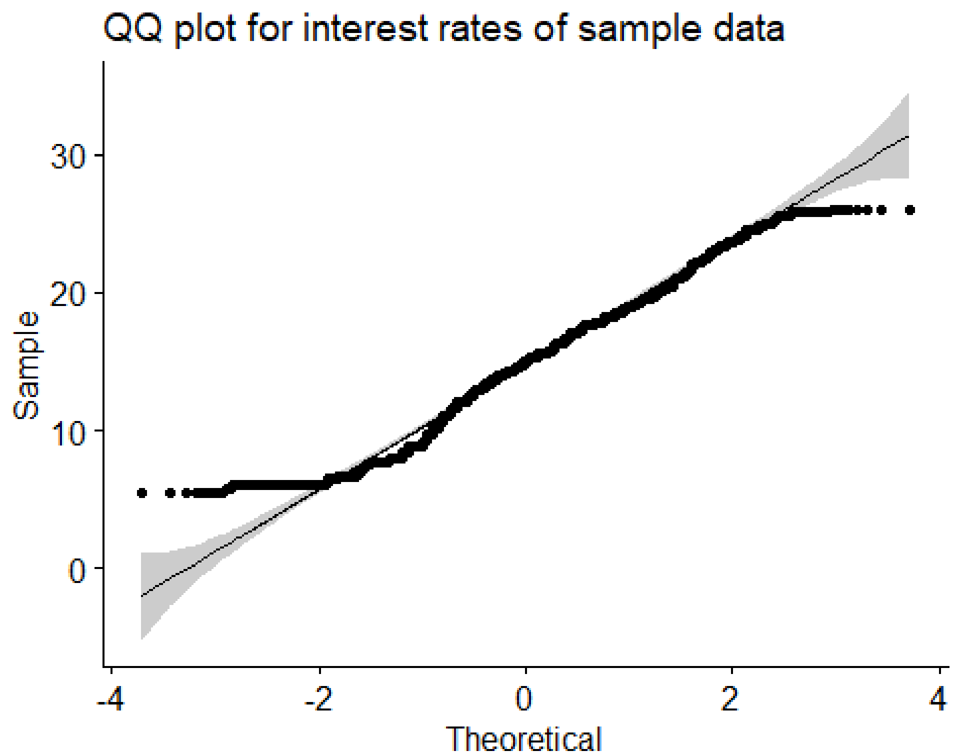
```
library(ggpubr)

## Loading required package: magrittr

##
## Attaching package: 'magrittr'

## The following object is masked from 'package:tidyr':
##
##     extract

## The following object is masked from 'package:purrr':
##
##     set_names

par(mfrow=c(2,2))
ggdensity(sample_data$X1, main="Density plot of interest rates", xlab="Intere
st Rate")
```



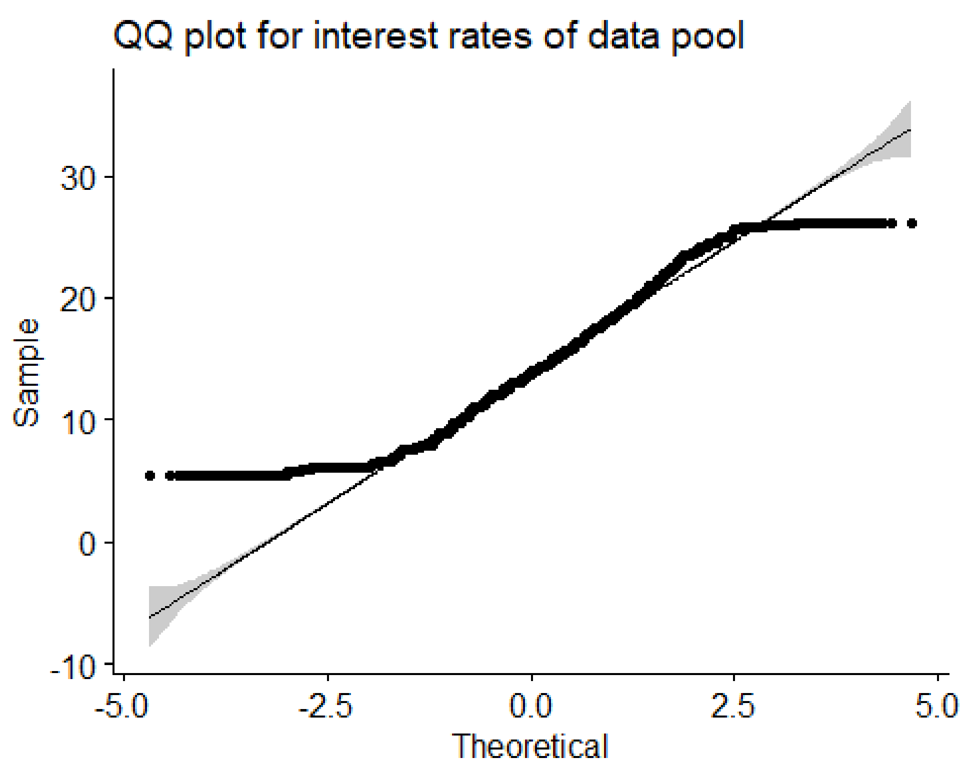

```r
ggdensity(load_data3$X1, main="Density plot of interest rates", xlab="Interes
t Rate")
```



```r
ggqqplot(sample_data$X1)+ggtitle("QQ plot for interest rates of sample data")
```

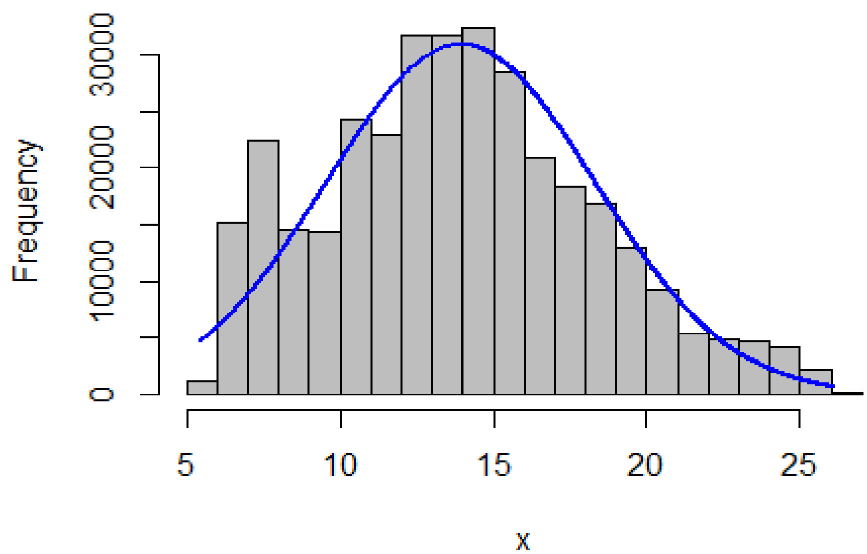



```r
ggqqplot(load_data3$X1)+ggtitle("QQ plot for interest rates of data pool")
```

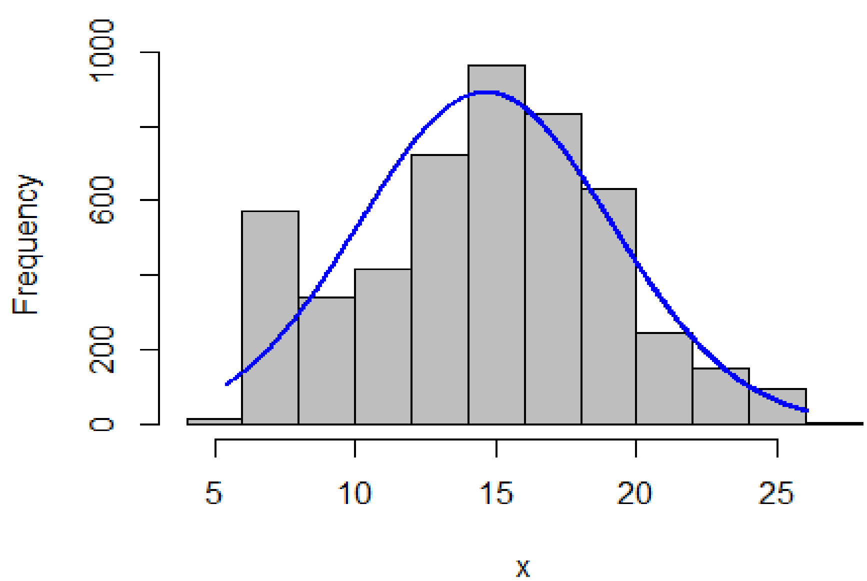


3.3 Normality of X1 (loan interest rates)

```
library(rcompanion)
plotNormalHistogram(load_data3$X1)
```




```
plotNormalHistogram(sample_data$X1)
```



4. Fit model

4.1 Split data to train and test data

Split data onto two parts, the ratio of training data to test data is 0.8:0.2

```
n=5000
ind=sample(c(TRUE, FALSE), n, replace=TRUE, prob=c(0.8, 0.2))
train=sample_data[ind,]
test=sample_data[!ind,]
train_y=train$X1
train_x=train[,-c(2)]
test_y=test$X1
test_x=test[, -c(2)]
```

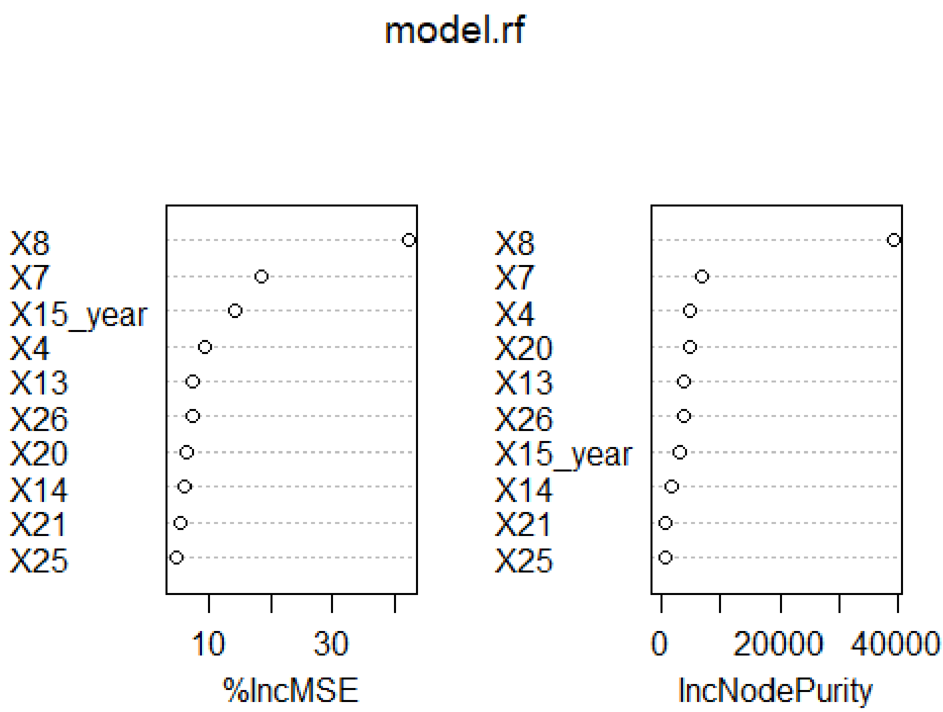
4.2 Random Forest

Using Random Forest to fit the data, get the test error is 6.48, and the most 5 important variables are X8, X7, X15_year, X4, and X14.

```
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:dplyr':
##
##     combine

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

set.seed(9933)
# fit model with random forest
model.rf=randomForest(X1~., data=sample_data,subset=ind, mtry=2,ntree=50, imp
ortance=T) # fit the random forest
predict.rf=predict(model.rf, newdata = test)
# Estimate test error rate
rf.se=mean((predict.rf-test_y)^2)
rf.se

## [1] 6.482732

#Get variable importance measure for each predictor
importance(model.rf)
```

```
##              %IncMSE IncNodePurity
## X8        42.319369    39157.5866
## X4         9.445326     4896.4772
## X13        7.479635     3943.8031
## X20        6.346545     4847.0800
## X21        5.421538      885.5191
## X25        4.677637      682.2733
## X26        7.439834     3730.6430
## X7        18.413410     6846.0579
## X14        6.160557     1880.5187
## X15_year 14.351519     3156.2364
```

```r
varImpPlot(model.rf)
```



4.3 Boosting

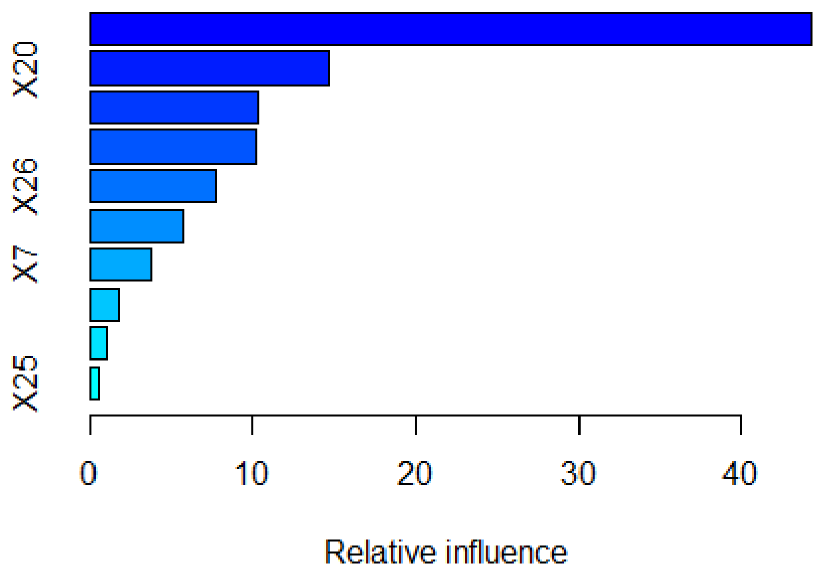
The boosted regression model has a mean error rate of 7.73. And the relative critical variables are X8, X20, X13, X4, X26, X15_year.

```r
library(gbm)
```

```
## Loaded gbm 2.1.5
```

```r
set.seed(9933)
 # Fit a boosted regression tree
model.boost=gbm(X1~ ., data = train, distribution = "gaussian",
    n.trees = 5000, interaction.depth = 4)
```

```r
# Get the relative influence plot
summary(model.boost)
```




```
##                 var    rel.inf
## X8               X8 44.3146240
## X20             X20 14.6785864
## X13             X13 10.3520500
## X4               X4 10.1777630
## X26             X26  7.7226799
## X15_year X15_year  5.7064901
## X7               X7  3.7765446
## X14             X14  1.7421007
## X21             X21  0.9802448
## X25             X25  0.5489165
```

```r
# Estimate test error rate for the boosted model
predict.boost <- predict(model.boost, newdata = test,
    n.trees = 5000)
boost.se=mean((predict.boost - test_y)^2)
boost.se

## [1] 7.727404
```

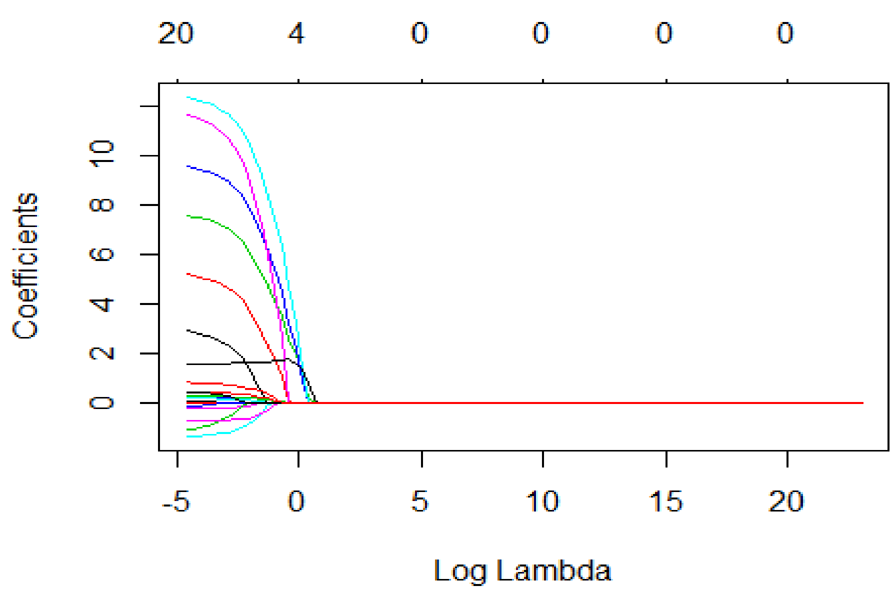
4.4 Lasso Regression

Lasso model shows the mean squared error is 6.62.

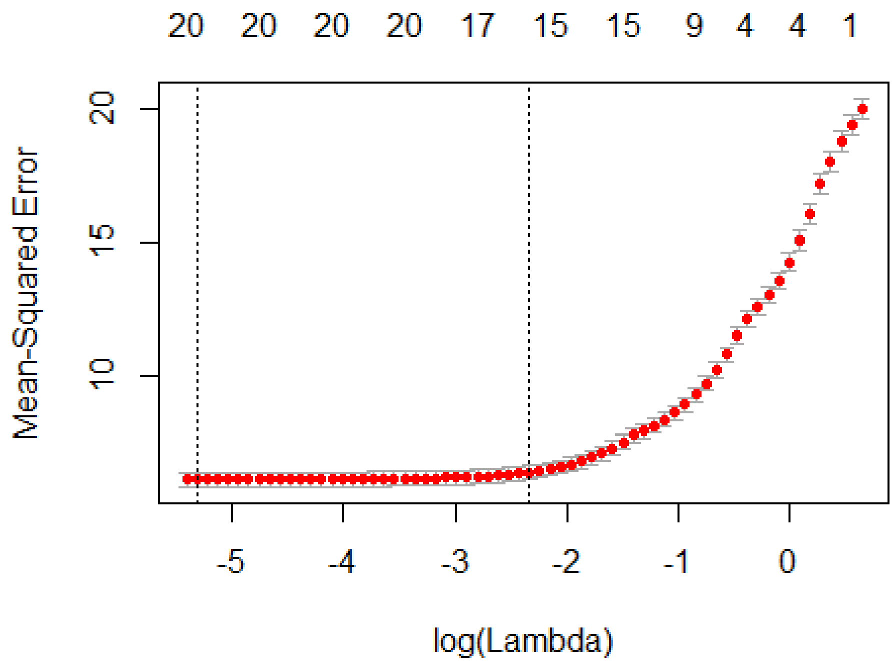
```r
library(glmnet)

## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'

## The following object is masked from 'package:tidyr':
##
##     expand

## Loading required package: foreach

##
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':
##
##     accumulate, when

## Loaded glmnet 2.0-16
```

```
set.seed(9933)
x=model.matrix(X1 ~ ., sample_data)[, -1]
y=sample_data$X1
# Set up a grid of lambda values (from 10^10 to 10^(-2)) in decreasing sequen
ce
grid <- 10^seq(10, -2, length = 100)
# fit lasso with each lambda
model.lasso <- glmnet(x[ind,], y[ind], alpha = 1, lambda = grid)
plot(model.lasso, xvar = "lambda")
```




```
# Use cross-validation to estimate test MSE using training data
cv.out <- cv.glmnet(x[ind,], y[ind], alpha = 1)
plot(cv.out)
```

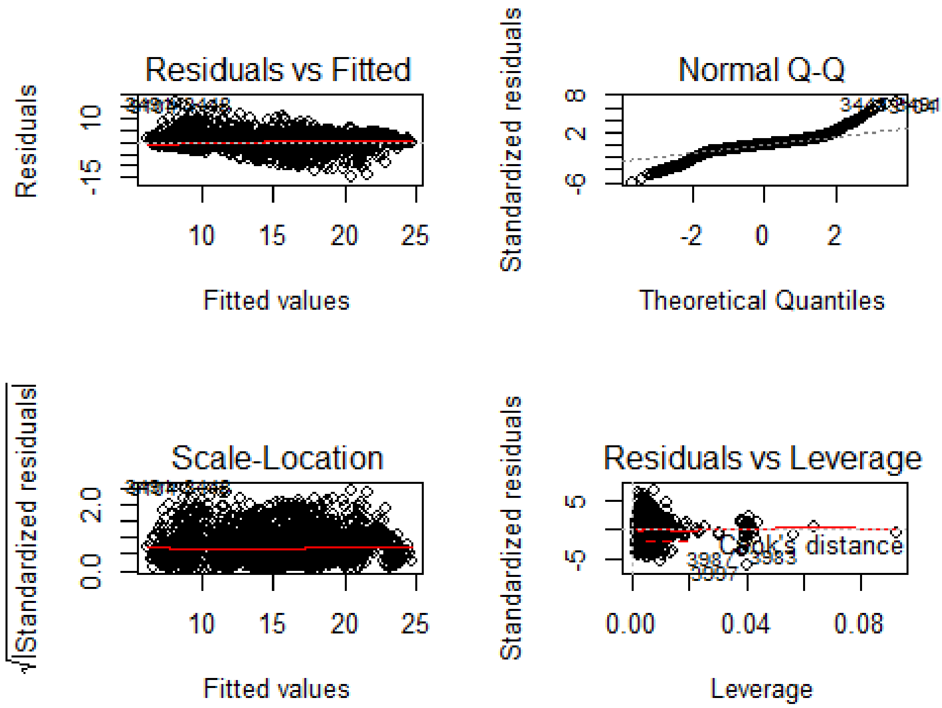



```
bestlam <- cv.out$lambda.min
bestlam

## [1] 0.004925728

predict.lasso <- predict(model.lasso, s = bestlam, newx = x[!ind,])
lasso.se=mean((predict.lasso - test_y)^2)
lasso.se

## [1] 6.622592
```

4.5 Linear Regression

First, select the model with the lowest AIC value.

```
model.final=lm(X1 ~ X7.60.months + X8D + X8E + X8F + X8C + X8G +
    X8B + X15_year13 + X15_year10 + X15_year11 + X14VERIFIED...income +
    X26 + X20 + X21 + X25 + X15_year12 + X15_year8, data = dummy_train)
par(mfrow=c(2,2))
plot(model.final)
```



Fit regression model and the mean squared error rate is 6.59.

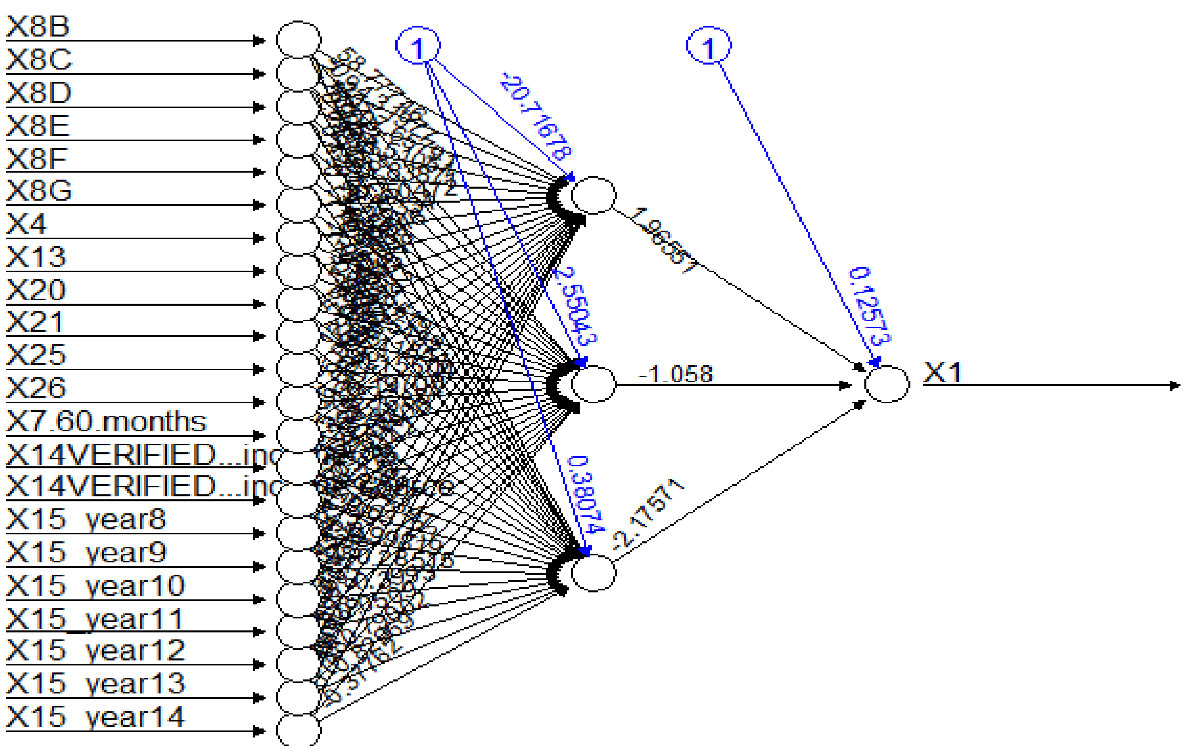
```
predict.lr=predict(model.final, newdata =dummy_test[,-c(1)] , se.fit = T, int
erval = "confidence")
lr.se=mean((predict.lr$fit[,1]-test_y)^2)
lr.se

## [1] 6.58692
```

4.6 Neural Network

Fit neural network model with 3 hidden layers. The mean squared error of the neural network model is 7.16.

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

##
## Attaching package: 'neuralnet'

## The following object is masked from 'package:dplyr':
##
##     compute

set.seed(9933)
#scale X1
scale_01 <- function(x){
  (x - min(x)) / (max(x) - min(x))
```

```
}
train.y=scale_01(train_y)
test.y=scale_01(test_y)

### create dummy variables
dummy_train_x=model.matrix(X1~., dat=train)[,-1]
dummy_test_x=model.matrix(X1~., data=test)[,-1]

dummy_train=data.frame(cbind(train.y,dummy_train_x))
names(dummy_train)[names(dummy_train)=="train.y"]="X1"

dummy_test=data.frame(cbind(test.y, dummy_test_x))
names(dummy_test)[names(dummy_test)=="test.y"]="X1"

set.seed(9933)
model.nn=neuralnet(X1~., data=dummy_train, hidden = 3, err.fct="sse",linear.o
utput = F)
plot(model.nn)
predict.nn=compute(model.nn, dummy_test[, -c(1)])

nn.se=sum((predict.nn$net.result-test.y)^2)/2
nn.se

## [1] 7.161134
```



5. Conclusion

Compare mean squared error from the previous five models, then the conclusion is that the Random Forest model has minimal MSE. To better know the accuracies of these models, resampling and test will be helpful.

```
model_name=c("Random Forest", "Boosting", "Lasso", "Linear Regression","Neura
l Network")
mse_value=c(rf.se, boost.se, lasso.se, lr.se, nn.se)
MSE=data.frame(model_name, mse_value)
MSE
```

```
##          model_name mse_value
## 1     Random Forest  6.482732
## 2          Boosting  7.727404
## 3             Lasso  6.622592
## 4 Linear Regression  6.586920
## 5    Neural Network  7.161134
```