MXN600 Major Data Anaysis Project

Joanna Salerno, Pavan Asopa, Sevin Nejadi, Fernanda Martins Giuriati

2023-10-07

Introduction

The credit risk models our lending start up company uses are of the utmost importance to the functioning and ultimate success of the company. As we were recently acquired by a regional Australian bank, the success of our company also impacts the success of the bank. Recently, some of the bank's senior financial analysts have raised concerns about the credit risk models we have been using. They reviewed our models' performance benchmarks and feel as though our models are not suitable for use in a setting in which they are subject to strict regulatory requirements.

Thus, we have been tasked by the bank's management to rebuild our credit risk model from the ground up. As per management, our main objective is to use information known at the time of a loan application to build a model that predicts loan default. We will follow a standard statistical analysis process, which will be guided by the following questions:

1. How does this new model perform compared to the one used previously? How can it be expected to perform on new loan applications?
2. What are the important variables in this model and how do they compare to variables that are traditionally important for predicting credit risk in the banking sector?

Furthermore, management has consulted with an expert statistician, who has suggested we also account for variation in trends that may exist either between different jurisdictions or over time. The following questions will guide this second part of our analysis:

3. Can accounting for this variation (e.g., state/zip-code and time) improve performance benchmarks?
4. Are there any surprising differences in variables that are important for predicting credit risk?
5. Does credit risk change over time or between states? This is not something the bank has previously investigated and results may inform modified loan policies in the future.

This report will document our entire analysis process, beginning with data exploration and cleaning, to model building and interpretations of our results.

Setup

We will first load in the required libraries for our data exploration and analysis process.

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
```

```
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
##
## Attaching package: 'MASS'
##
##
## The following object is masked from 'package:dplyr':
##
##     select
##
##
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
##
## Loading required package: Matrix
##
##
## Attaching package: 'Matrix'
##
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
##
##
## Attaching package: 'lmerTest'
##
##
## The following object is masked from 'package:lme4':
##
##     lmer
##
##
## The following object is masked from 'package:stats':
##
##     step
```

Next, we will load in our datasets. We have a total of 4:

(1) A training dataset that we will use to build and train our model
(2) A test dataset that we will use to test the fit of our model(s)
(3) A validation dataset that we will use to assess the performance of our model(s)
(4) An extended dataset that includes the necessary variables for us to account for variation such as location and time

```r
train_data <- read.csv("benchmark_training_loan_data.csv")
test_data <- read.csv("benchmark_testing_loan_data.csv")
```

```r
val_data <- read.csv("benchmark_validation_loan_data.csv")
extended <- read.csv("extendend_version_loan_data.csv")
```

Exploratory Analysis

We will begin by exploring the available data to understand how each variable is distributed and to identify any potential data quality issues. We will also investigate the relationships between the different variables to see whether any variables are highly correlated with one another.

Note: For this exploration portion of our analysis, we will be using the training dataset.

We will first explore the training dataset to understand its structure and the variables it is comprised of.

```r
head(train_data)
```

```
##   X loan_amnt      term int_rate emp_length home_ownership annual_inc
## 1 1      2500 36 months    13.98    4 years           RENT      20004
## 2 2      5000 36 months    15.95    4 years           RENT      59000
## 3 3      7000 36 months     9.91  10+ years       MORTGAGE      53796
## 4 4      2000 36 months     5.42  10+ years           RENT      30000
## 5 5      8000 36 months     6.03        n/a       MORTGAGE      77736
## 6 6      6250 36 months    17.27    4 years       MORTGAGE      28000
##   verification_status            purpose   dti delinq_2yrs inq_last_6mths
## 1        Not Verified              other 19.86           0              5
## 2        Not Verified debt_consolidation 19.57           0              1
## 3        Not Verified              other 10.80           3              3
## 4        Not Verified debt_consolidation  3.60           0              0
## 5            Verified              other  6.07           0              0
## 6            Verified              other 13.76           0              0
##   open_acc pub_rec revol_bal revol_util total_acc repay_fail credit_age_yrs
## 1        7       0       981     0.2130        10          0       4.931319
## 2        7       0     18773     0.9990        15          1      16.222527
## 3        7       0      3269     0.4720        20          0      13.549451
## 4        7       0         0     0.0000        15          0      36.791209
## 5       12       0      4182     0.1360        49          0      15.302198
## 6        2       1         0     0.0846        15          1      12.126374
```

At first glance of the first 6 rows of this dataset, we notice there is a value of n/a in the employment length column. There are also a few zero values in a few columns. This will prompt us to further explore the data for any true missing values.

```r
dim(train_data)
```

```
## [1] 23052    19
```

The training dataset contains a total of 23,052 observations of 19 variables.

```r
str(train_data)
```

```
## 'data.frame':    23052 obs. of  19 variables:
##  $ X                  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ loan_amnt          : int  2500 5000 7000 2000 8000 6250 8000 16000 7000 13000 ...
##  $ term               : chr  "36 months" "36 months" "36 months" "36 months" ...
##  $ int_rate           : num  13.98 15.95 9.91 5.42 6.03 ...
##  $ emp_length         : chr  "4 years" "4 years" "10+ years" "10+ years" ...
##  $ home_ownership     : chr  "RENT" "RENT" "MORTGAGE" "RENT" ...
##  $ annual_inc         : num  20004 59000 53796 30000 77736 ...
##  $ verification_status: chr  "Not Verified" "Not Verified" "Not Verified" "Not Verified" ...
##  $ purpose            : chr  "other" "debt_consolidation" "other" "debt_consolidation" ...
##  $ dti                : num  19.86 19.57 10.8 3.6 6.07 ...
##  $ delinq_2yrs        : int  0 0 3 0 0 0 0 0 2 0 ...
##  $ inq_last_6mths     : int  5 1 3 0 0 0 1 0 1 1 ...
##  $ open_acc           : int  7 7 7 7 12 2 8 5 3 14 ...
##  $ pub_rec            : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ revol_bal          : int  981 18773 3269 0 4182 0 9287 11006 6082 38433 ...
##  $ revol_util         : num  0.213 0.999 0.472 0 0.136 0.0846 0.619 0.651 0.965 0.565 ...
##  $ total_acc          : int  10 15 20 15 49 15 37 36 11 31 ...
##  $ repay_fail         : int  0 1 0 0 0 1 0 0 0 0 ...
##  $ credit_age_yrs     : num  4.93 16.22 13.55 36.79 15.3 ...
```

Upon further investigation of the structure of the training data, we can see that 14 of the 19 variables are currently of numeric type, and the remaining 5 variables are characters. We will now further investigate each variable to determine whether there is any missing data or outliers.

```r
summary(train_data)
```

```
##        X           loan_amnt         term              int_rate    
##  Min.   :    1   Min.   :  500   Length:23052       Min.   : 5.42  
##  1st Qu.: 5764   1st Qu.: 5400   Class :character   1st Qu.: 9.63  
##  Median :11526   Median : 9862   Mode  :character   Median :11.99  
##  Mean   :11526   Mean   :11128                      Mean   :12.18  
##  3rd Qu.:17289   3rd Qu.:15000                      3rd Qu.:14.72  
##  Max.   :23052   Max.   :35000                      Max.   :24.11  
##   emp_length        home_ownership       annual_inc      verification_status
##  Length:23052       Length:23052       Min.   :   2000   Length:23052       
##  Class :character   Class :character   1st Qu.:  40032   Class :character   
##  Mode  :character   Mode  :character   Median :  58000   Mode  :character   
##                                        Mean   :  68435                      
##                                        3rd Qu.:  82000                      
##                                        Max.   :2039784                      
##    purpose               dti          delinq_2yrs     inq_last_6mths 
##  Length:23052       Min.   : 0.000   Min.   :0.0000   Min.   : 0.000  
##  Class :character   1st Qu.: 8.248   1st Qu.:0.0000   1st Qu.: 0.000  
##  Mode  :character   Median :13.550   Median :0.0000   Median : 1.000  
##                     Mean   :13.426   Mean   :0.1502   Mean   : 1.078  
##                     3rd Qu.:18.740   3rd Qu.:0.0000   3rd Qu.: 2.000  
##                     Max.   :29.950   Max.   :9.0000   Max.   :33.000  
##     open_acc         pub_rec        revol_bal        revol_util    
##  Min.   : 1.000   Min.   :0.000   Min.   :     0   Min.   :0.0000  
##  1st Qu.: 6.000   1st Qu.:0.000   1st Qu.:  3686   1st Qu.:0.2600  
##  Median : 9.000   Median :0.000   Median :  8918   Median :0.4990  
##  Mean   : 9.354   Mean   :0.057   Mean   : 14383   Mean   :0.4932  
```

```
##  3rd Qu.:12.000   3rd Qu.:0.000   3rd Qu.: 17367   3rd Qu.:0.7302
##  Max.   :47.000   Max.   :4.000   Max.   :952013   Max.   :1.1900
##    total_acc       repay_fail    credit_age_yrs
##  Min.   : 1.00   Min.   :0.000   Min.   : 0.5055
##  1st Qu.:13.00   1st Qu.:0.000   1st Qu.: 9.0302
##  Median :20.00   Median :0.000   Median :12.5440
##  Mean   :22.12   Mean   :0.152   Mean   :13.7856
##  3rd Qu.:29.00   3rd Qu.:0.000   3rd Qu.:17.1429
##  Max.   :90.00   Max.   :1.000   Max.   :60.6209
```

Included above is the numeric distributions of each of the included variables. Based on the above summary, it appears as though there may exist one or multiple outliers in a few variables: annual income, inquiries in the last 6 months, open accounts, revolving balances, and total accounts. However, we will need to further investigate the distribution of all variables to better visualize this to determine whether these actually appear to be outliers.

Before producing some exploratory plots, we will briefly explore the data to see whether there are missing values for us to handle.

```r
colSums(is.na(train_data))
```

```
##                   X           loan_amnt                term            int_rate
##                   0                   0                   0                   0
##          emp_length      home_ownership          annual_inc verification_status
##                   0                   0                   0                   0
##             purpose                 dti         delinq_2yrs      inq_last_6mths
##                   0                   0                   0                   0
##            open_acc             pub_rec           revol_bal          revol_util
##                   0                   0                   0                   0
##           total_acc          repay_fail      credit_age_yrs
##                   0                   0                   0
```

Based on the above output, it appears as though this dataset does not contain any missing data in the form of NA values. Now, we will explore the n/a values present in the columns which include string data.

```r
sum(train_data == 'n/a')
```

```
## [1] 591
```

```r
na_rows <- train_data %>% filter_all(any_vars(. %in% c('n/a')))
head(na_rows)
```

```
##     X loan_amnt      term int_rate emp_length home_ownership annual_inc
## 1   5      8000 36 months     6.03        n/a       MORTGAGE      77736
## 2  25      1450 36 months     7.51        n/a           RENT      10000
## 3  43      1800 36 months     5.42        n/a            OWN      29184
## 4 100      4000 36 months    17.19        n/a            OWN      37200
## 5 131      2250 36 months     5.42        n/a           RENT      52500
## 6 155      5500 36 months     8.49        n/a       MORTGAGE      36780
##   verification_status            purpose   dti delinq_2yrs inq_last_6mths
## 1            Verified              other  6.07           0              0
## 2     Source Verified              other 22.20           0              0
```

```
## 3        Not Verified     small_business 23.68           0              2
## 4        Not Verified   home_improvement  9.35           0              0
## 5        Not Verified debt_consolidation 16.43           0              0
## 6        Not Verified                car 16.54           0              2
##   open_acc pub_rec revol_bal revol_util total_acc repay_fail credit_age_yrs
## 1       12       0      4182      0.136        49          0      15.302198
## 2        9       0       709      0.032        10          0       6.271978
## 3        7       0     30037      0.235        24          0      36.711538
## 4        6       0      4598      0.575        14          1      12.881868
## 5        9       0      4425      0.360        25          0      18.978022
## 6        8       0      8926      0.498        27          1      31.857143
```

Based on the above output, there are a total of 591 n/a (string) in this dataset. These all appear to be from the employment length column. We will move forward and assume that this is not truly missing data, but rather, means that the applicant is not currently employed. We will further examine the ditribution of each variable by creating exploratory plots, and this will further clarify the meaning of some of the values contained within each column of the dataset.

Exploratory Plots

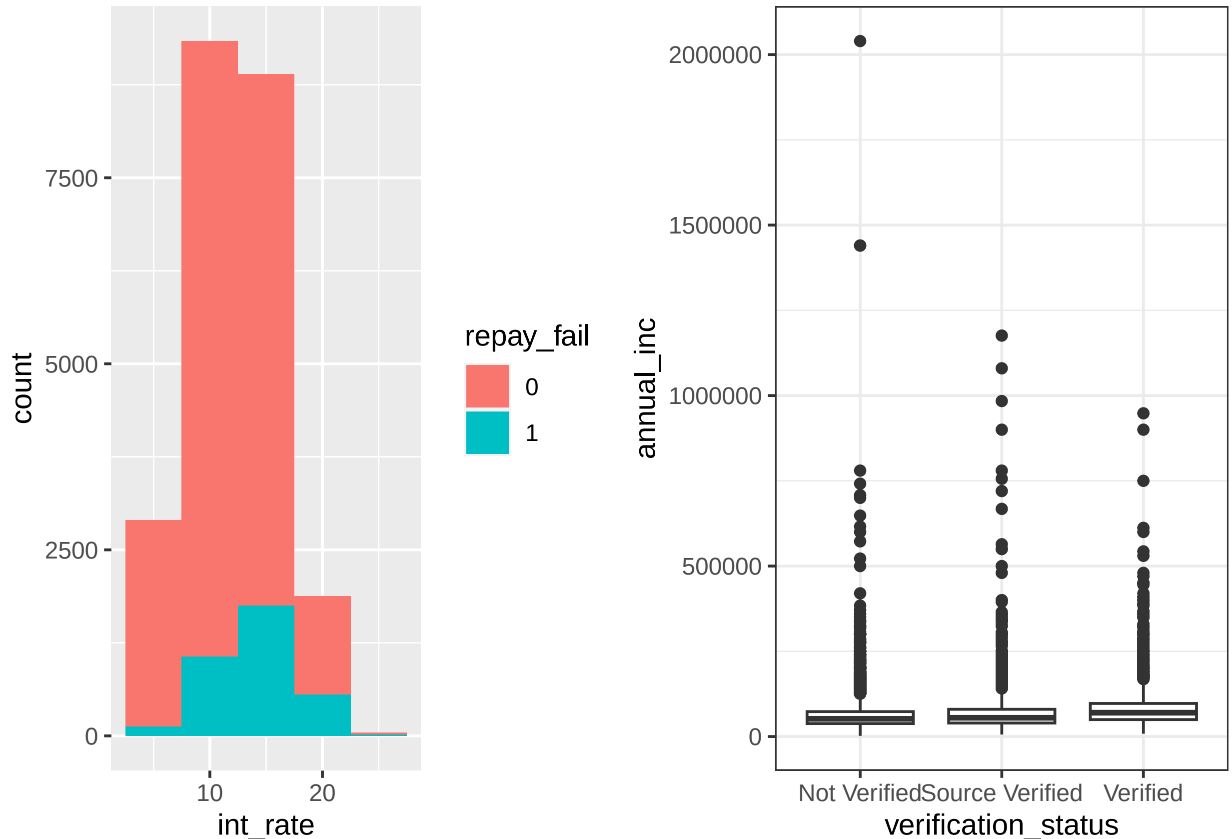
Before we can create exploratory plots, there are a few variables we will need to convert to factors. This will help us in creating our visualizations as well as conducting our analyses, so we can consider the selected variables as factors with different levels rather than simply strings.

```r
train_data$term <- as.factor(train_data$term)
train_data$emp_length <- as.factor(train_data$emp_length)
train_data$home_ownership <- as.factor(train_data$home_ownership)
train_data$verification_status <- as.factor(train_data$verification_status)
train_data$purpose <- as.factor(train_data$purpose)
train_data$repay_fail <- as.factor(train_data$repay_fail)
```

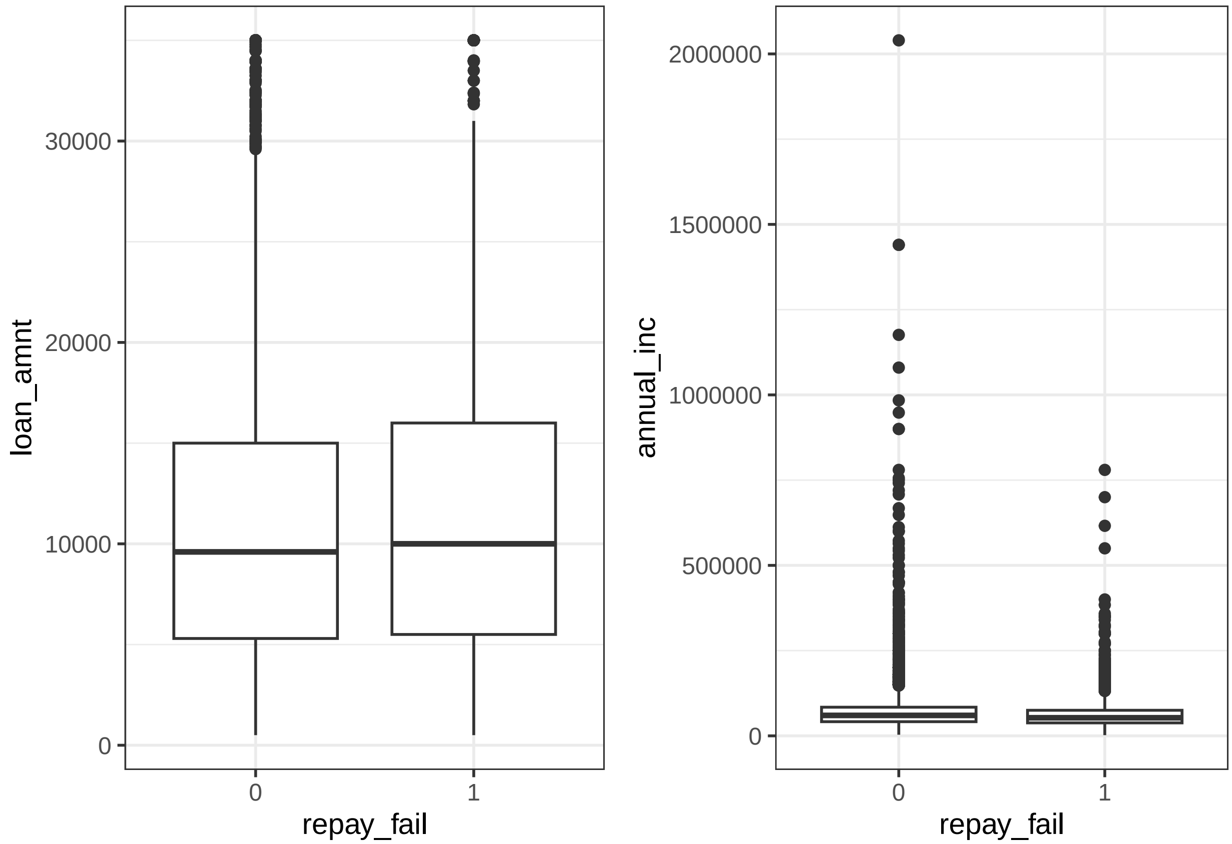
Now, we will create some histograms to visualize the distributions of some of the string/character variables in this dataset.

```r
p1 <- ggplot(data = train_data, aes(int_rate, fill = repay_fail)) +
  geom_histogram(binwidth=5)

p2 <- ggplot(data = train_data, mapping = aes(x = verification_status, y = annual_inc)) +
  geom_boxplot() +
  theme_bw()

p3 <- ggplot(data = train_data, mapping = aes(x = repay_fail, y = loan_amnt)) +
  geom_boxplot() +
  theme_bw()

p4 <- ggplot(data = train_data, mapping = aes(x = repay_fail, y = annual_inc)) +
  geom_boxplot() +
  theme_bw()

p5 <- ggplot(data = train_data, mapping = aes(x = annual_inc, y = loan_amnt)) +
  geom_point() +
  theme_bw()
```

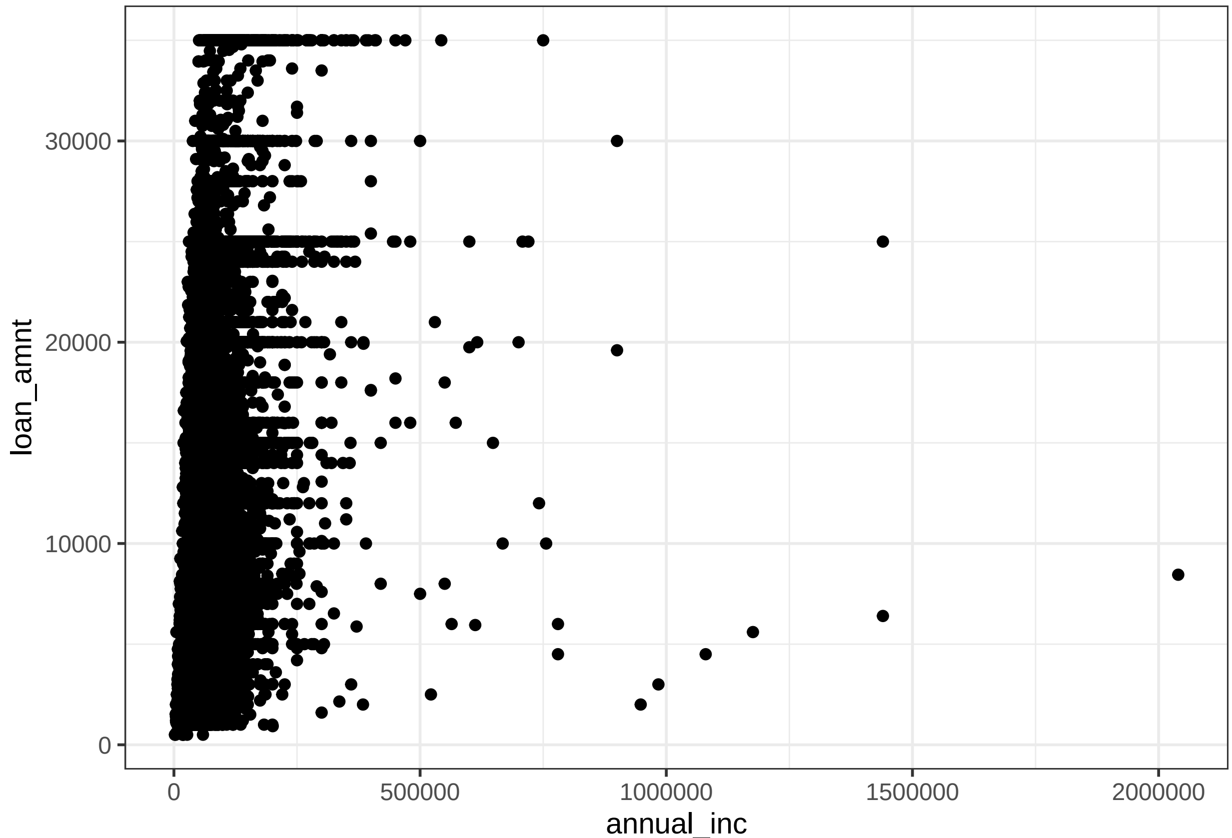
```r
ggarrange(p1, p2)
```




```r
ggarrange(p3, p4)
```

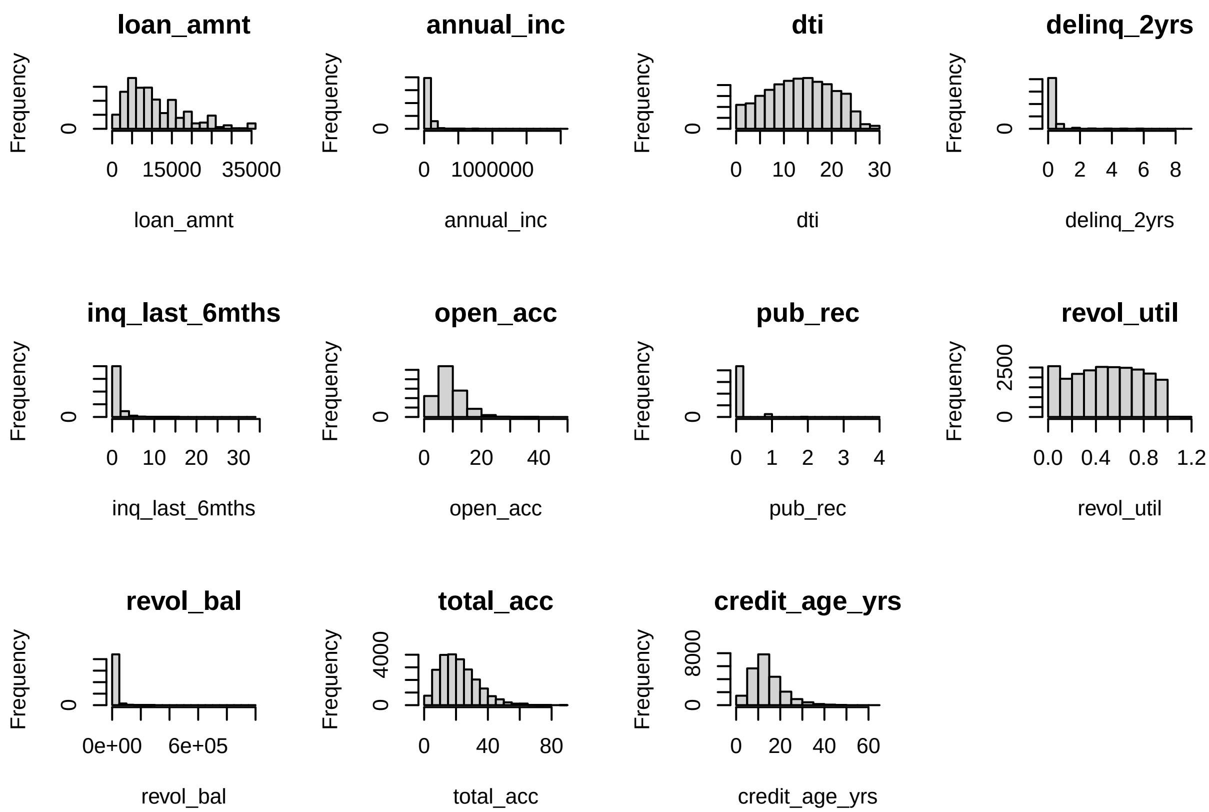



```
ggarrange(p5)
```

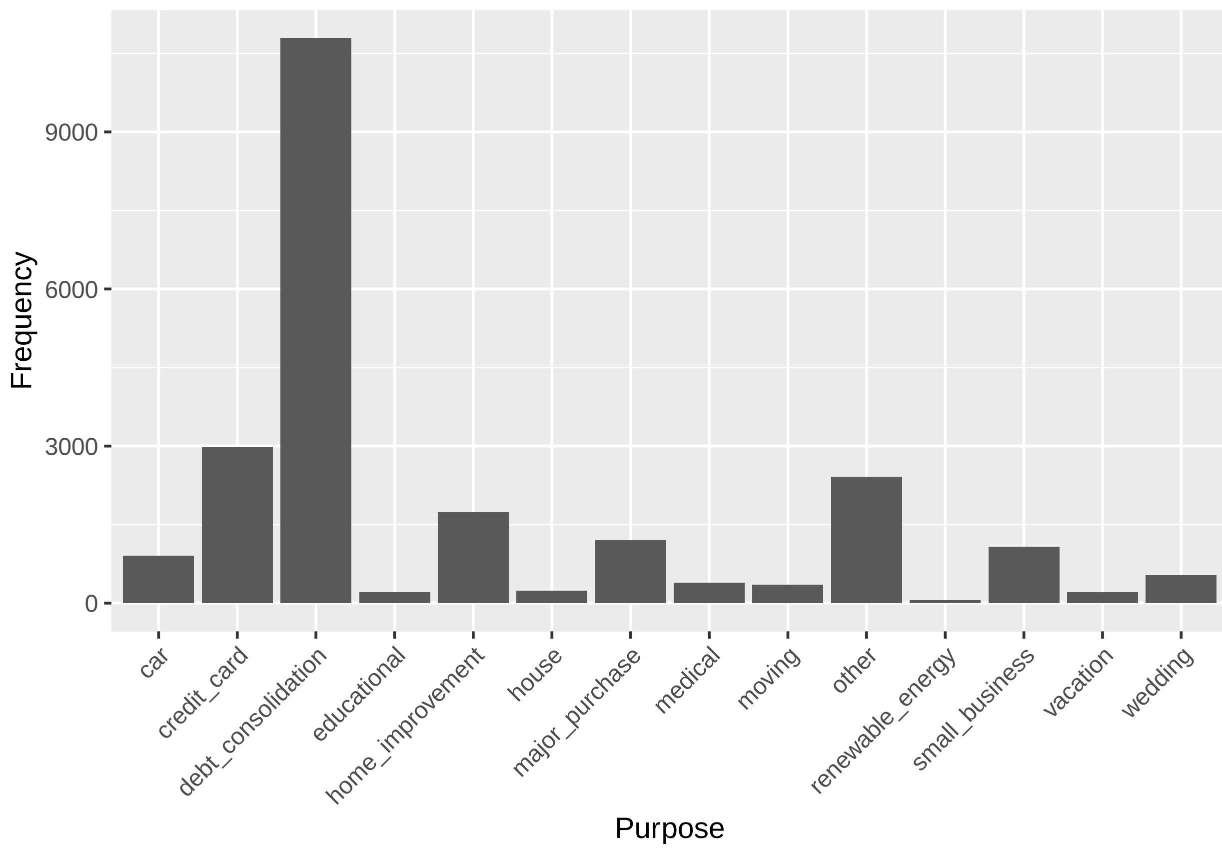


Here we create some distribution plots to have a better visualization of the data:

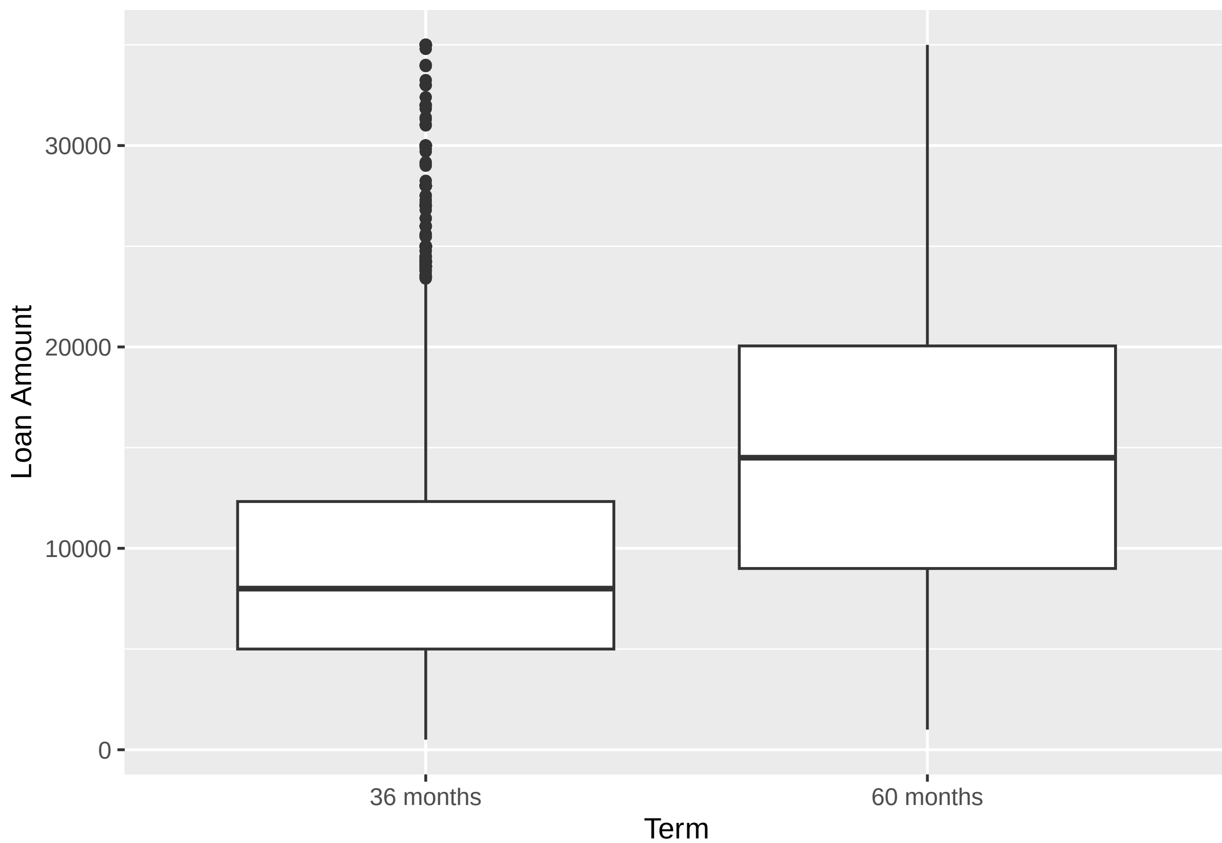
```
numeric_vars <- c("loan_amnt", "annual_inc", "dti", "delinq_2yrs", "inq_last_6mths",
                  "open_acc", "pub_rec", "revol_util", "revol_bal", "total_acc",
                  "credit_age_yrs")
par(mfrow=c(3,4))
for (var in numeric_vars) {
  hist(train_data[[var]], main=var, xlab=var)
}
```




```r
ggplot(train_data, aes(x = purpose)) +
  geom_bar() +
  labs(x = "Purpose", y = "Frequency") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
ggplot(train_data, aes(x = term, y = loan_amnt)) +
  geom_boxplot() +
  labs(x = "Term", y = "Loan Amount")
```



*** EXPLAIN PLOTS ***

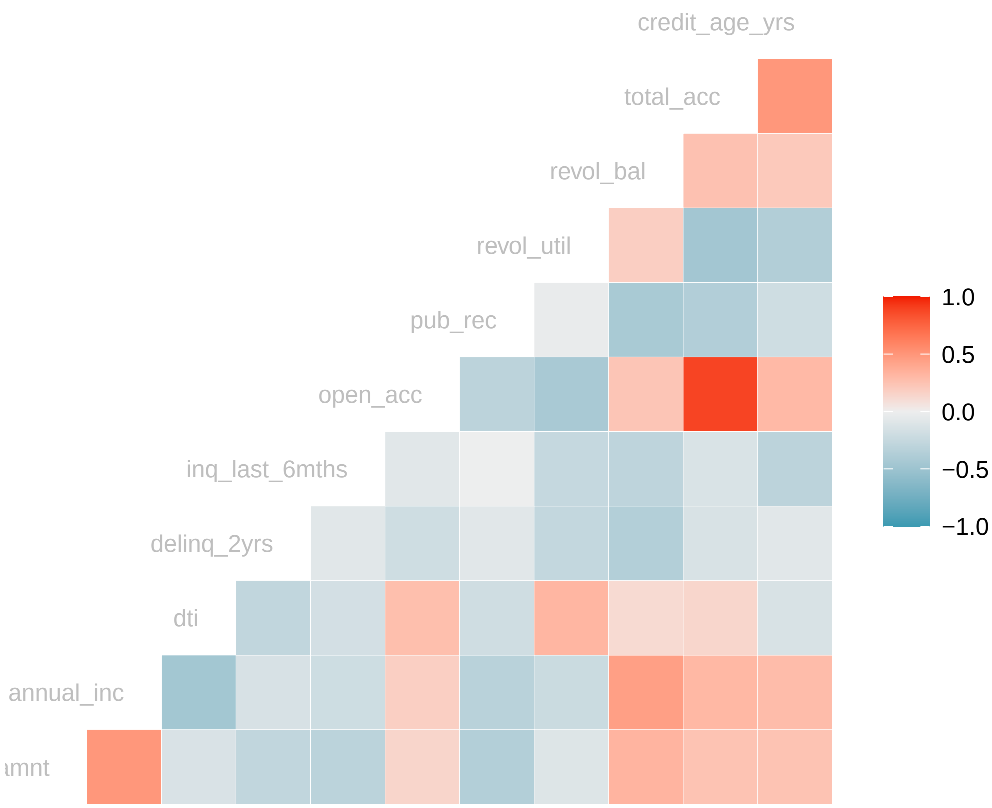
Now, we create a plot that depicts the correlation between the different numeric variables; it includes both correlation coefficients and visuals of the distribution of each included variable.

```
# create ggpairs plot with the specified columns
ggpairs_plot <- ggpairs(train_data[, c("loan_amnt","int_rate", "annual_inc", "dti",
                                      "delinq_2yrs", "inq_last_6mths", "open_acc",
                                      "pub_rec", "revol_util", "revol_bal", "total_acc",
                                      "credit_age_yrs")],
                        progress = FALSE)

#ggpairs_plot
```

Below, we create a correlation matrix of the numeric variables to help us understand the relationship between the different numeric variables.

```
columns <- c("loan_amnt", "annual_inc", "dti", "delinq_2yrs", "inq_last_6mths", "open_acc", "pub_rec",
corr_matrix <- cor(train_data[columns], method = "pearson")
ggcorr(corr_matrix, type = "lower", hjust = 1, size = 3, color = "grey")
```

```
## Warning in geom_text(data = textData, aes_string(label = "diagLabel"), ..., :
## Ignoring unknown parameters: ‘type‘
```



New Credit Risk Model

```
# #Full model with all possible interactions for backwards selection:
# full_interaction_model <- glm(data = train_data,
#     formula = repay_fail ~ loan_amnt* term * int_rate * emp_length * home_ownership * annual_inc *
#       verification_status * purpose * dti * delinq_2yrs * inq_last_6mths * open_acc * pub_rec *
#       revol_bal * revol_util * total_acc * credit_age_yrs,
#     family = "binomial")
#
# #Model with no variables present for forwards selection:
# null_model <- glm(data = train_data,
#     formula = repay_fail ~ 1,
#     family = "binomial")
#
# #Perform backward and forward selection:
# backward_sel_model <- stepAIC(
#   full_interaction_model, direction = "backward",trace = 0)
# forward_sel_model <- stepAIC(
#   null_model,
#   scope = formula(full_interaction_model),
#   direction = "forward",
#   trace = 0) ## trace = 0 prevents automatic output of stepAIC function.
```

Extended Credit Risk Model

Conclusion

Limitations

Future Directions