Lending Club - Data Preparation

Constantin Romanescu

October 30, 2018

Table of Contents

[Set up the R environment 2](#_Toc529564252)

[1. Data Collection and Business Understanding 3](#_Toc529564253)

[Load data 3](#_Toc529564254)

[Change feature data formats 5](#_Toc529564255)

[Loan counts and default rates by state 7](#_Toc529564256)

[Time-dependence of the loan status. Observations selection 10](#_Toc529564257)

[2. Data Pre-processing 14](#_Toc529564258)

[Additional feature selection 14](#_Toc529564259)

[Add features with socio-economic data 16](#_Toc529564260)

[Group states by region 16](#_Toc529564261)

[Split the data into train - validation - test sets 17](#_Toc529564262)

[Missing data imputation 17](#_Toc529564263)

[Analysis of the numeric independent in the data 18](#_Toc529564264)

[Remove numeric variables that are highly correlated 18](#_Toc529564265)

[Numeric data summary 20](#_Toc529564266)

[Numerical feature description 21](#_Toc529564267)

[1. Loan amount (loan\_amnt) 21](#_Toc529564268)

[2. Interest rate (int\_rate) 22](#_Toc529564269)

[3. Annual income (annual\_inc) 23](#_Toc529564270)

[4. Debt-to-Income ratio (dti) 24](#_Toc529564271)

[5. Total accounts (total\_acc) 25](#_Toc529564272)

[6. Public records (pub\_rec) 26](#_Toc529564273)

[7. Revolving balance (revol\_bal) 27](#_Toc529564274)

[8. Revolving utilization ratio (revol\_util) 28](#_Toc529564275)

[9. Delinquencies in the last two years (delinq\_2yrs) 29](#_Toc529564276)

[10. Lentgh of credit history (Credit\_history) 30](#_Toc529564277)

[11. S&P500 monthly average (sp500) 31](#_Toc529564278)

[12. Treasury Bills 3 years interest (X3\_Yr\_avg) 32](#_Toc529564279)

[13. Monthly credit card interest rate (CreditCardAvg) 33](#_Toc529564280)

[14. Unemployment rate, by state and month-year (Unemp\_rate) 34](#_Toc529564281)

[Analysis of the categorical (factor) independent variables 35](#_Toc529564282)

[Variable independence 35](#_Toc529564283)

[Categorical (factor) variables 37](#_Toc529564284)

[1. Loan grade (grade) 37](#_Toc529564285)

[2. Loan sub-grade (sub\_grade) 39](#_Toc529564286)

[3. Loan purpose (purpose) 40](#_Toc529564287)

[4. Verification status (verification\_status) 42](#_Toc529564288)

[5. Home ownership (home\_ownership) 43](#_Toc529564289)

[6. Employment length (emp\_length) 45](#_Toc529564290)

[7. State (State) 47](#_Toc529564291)

[8. Geographical area (Area) 49](#_Toc529564292)

[Save the final files for the modelling phase 50](#_Toc529564293)

[Further feature selection using the Boruta package 50](#_Toc529564294)

[References 52](#_Toc529564295)

# Set up the R environment

knitr::opts\_chunk$set(echo = TRUE)  
  
# Install the R packages for the project  
packages = c('ggplot2', 'tidyverse', 'caret', 'caTools', 'GGaly', 'dplyr', 'readxl', 'lubridate', 'knitr',  
 'devtools', 'mapproj', 'gridExtra', 'moments', 'nortest', 'Boruta', 'CORElearn')  
  
for (package in packages){  
 if(!(package %in% installed.packages()[,"Package"])) {  
 install.packages(package, repos = "http://cran.rstudio.com/")  
 }  
}  
  
if(!('fiftystater' %in% installed.packages()[,"Package"])) {  
 devtools::install\_github("wmurphyrd/fiftystater")  
 }  
  
library(tidyverse)  
library(ggplot2)  
library(dplyr)  
library(caret)  
library(caTools)  
library(readxl)  
library(lubridate)  
library(knitr)  
library(fiftystater)  
library(mapproj)  
library(gridExtra)  
library(corrplot)  
library(moments)  
library(nortest)  
library(Boruta)  
library(CORElearn)

# 1. Data Collection and Business Understanding

## Load data

Lending Club publishes the data for the loans they have offered since 2007. The data is stored in smaller files, grouped by the date the transactions occured. As of October, 2018, the data includes the loans issued up to the second quarter of 2018, in 14 csv files at <https://www.lendingclub.com/info/download-data.action>. A description of the data features is available as an excel file at <https://resources.lendingclub.com/LCDataDictionary.xlsx>.

Since it is not possible to read the data directly from the web through an API without a Lending Club account, we manually downloaded the files, combined them into a single data frame, and stored it on the local disk (data\_2007\_2018Q2.csv).

if(file.exists("C:\\Ryerson Capstone\\data\\LC\\data\_2007\_2018Q2.csv")) {  
 data <- read\_csv("C:\\Ryerson Capstone\\data\\LC\\data\_2007\_2018Q2.csv")  
} else {  
 # choose file from a different location  
 data <- read\_csv(file = file.choose())  
}  
  
# Updated data for 2015 released in Nov. 2018  
if(file.exists("C:\\Ryerson Capstone\\data\\LC\\LC\_2015\_Nov2018.csv")) {  
 data\_2015\_updated <- read\_csv("C:\\Ryerson Capstone\\data\\LC\\LC\_2015\_Nov2018.csv")  
} else {  
 # choose file from a different location  
 data\_2015\_updated <- read\_csv(file = file.choose())  
}

## Warning: 2 parsing failures.  
## row # A tibble: 2 x 5 col row col expected actual file expected <int> <chr> <chr> <chr> <chr> actual 1 108731 funded\_am~ no trailing ch~ .111979~ "'C:\\Ryerson Capstone\\data~ file 2 323509 funded\_am~ no trailing ch~ .032087~ "'C:\\Ryerson Capstone\\data~

The dataset consists of 2004062 observations and 145 features. Our primary target variable is *loan\_status*. Later we will also look at the total payments received for each loan, encoded as *total\_pymnt*. Not all the features are available at the time a loan is listed. Some features are added later to keep track of the loan repayment history and to add or update features in borrower’s profile. Due to the sheer volume of the data, we defer data summarization and visualization to later sections.

An important note is that 6 features from the loan description are missing from our data. These features contain ranges of FICO scores for the primary and secondary applicants. Although these are important features in predicting the payment of a loan, they are also correlated with the Lending Club rating and the interest rate assigned to each loan. Complete datasets are available on Kaggle. (<https://www.kaggle.com/wordsforthewise/lending-club>)

Let’s look at the distribution of the target variable, **loan\_status**.

loan\_status\_count <- data %>%  
 select(loan\_status) %>%  
 group\_by(loan\_status) %>%  
 summarize(Count = n()) %>%  
 mutate(Percentage = round(100\* Count/nrow(data), 3)) %>%   
 arrange(desc(Count)) %>%  
 ungroup()  
  
knitr::kable(loan\_status\_count, caption = 'Table 1: Loan status')

Table 1: Loan status

|  |  |  |
| --- | --- | --- |
| loan\_status | Count | Percentage |
| Fully Paid | 873920 | 43.607 |
| Current | 865418 | 43.183 |
| Charged Off | 220712 | 11.013 |
| Late (31-120 days) | 21092 | 1.052 |
| In Grace Period | 14370 | 0.717 |
| Late (16-30 days) | 5766 | 0.288 |
| Does not meet the credit policy. Status:Fully Paid | 1988 | 0.099 |
| Does not meet the credit policy. Status:Charged Off | 761 | 0.038 |
| Default | 35 | 0.002 |

Not suprisingly, due to the boom in the P2P lending in the recent years, a large fraction of the loans are current loans. There are a number of loans that are late (~ 2%), that can also be classified as current loans. Note that the default category refers to loans that are late (120 - 150 days). A loan is classified as *Charged Off* if an installment is not payed after 150 days from the payment due date.

Based on the data description and literature studies (Cohen, Guetta, Jiao, & Provost, 2018), we define a list of features that are available to investors at the time a loan is listed, plus some features that will be use as target variables (*loan-Status*, *total\_pymnt*) or help in interpreting the dynamics of loan payment, *last\_pymnt\_d*. For now we’ll keep only the loans in the first three categories.

features\_of\_interest <- c('loan\_amnt', 'term', 'int\_rate', 'installment', 'grade', 'sub\_grade', 'purpose',  
 'verification\_status', 'home\_ownership', 'emp\_title', 'emp\_length', 'zip\_code', 'addr\_state',  
 'annual\_inc', 'dti', 'dti\_joint', 'annual\_inc\_joint', 'earliest\_cr\_line', 'open\_acc',  
 'total\_acc', 'num\_sats', 'pub\_rec', 'revol\_bal', 'revol\_util', 'delinq\_2yrs', 'recoveries',   
 'mths\_since\_last\_delinq', 'mths\_since\_last\_major\_derog', 'collections\_12\_mths\_ex\_med',   
 'mths\_since\_last\_record', 'issue\_d', 'loan\_status', 'total\_pymnt', 'last\_pymnt\_d')  
  
data\_preproc <- data %>% select(features\_of\_interest) %>%  
 filter(loan\_status %in% c('Fully Paid', 'Charged Off', 'Current'))  
new\_data\_preproc <- data\_2015\_updated %>% select(features\_of\_interest) %>%  
 filter(loan\_status %in% c('Fully Paid', 'Charged Off', 'Current'))  
  
# glimpse(data\_preproc)

## Change feature data formats

Now we’ll convert the features that contain dates into Date format and the features that contain percent data (*int\_rate*, *revol\_util*) into numeric format. Also, we introduce two new variables: length of credit history and the number of month a loan has been ion the book.

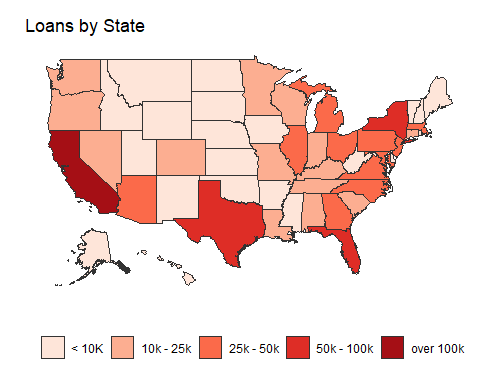
data\_preproc <- data\_preproc %>%  
 mutate(issue\_d = dmy(paste0('01-', issue\_d))) %>%  
 mutate(earliest\_cr\_line = dmy(paste0('01-', earliest\_cr\_line))) %>%  
 mutate(last\_pymnt\_d = dmy(paste0('01-', last\_pymnt\_d)))  
  
new\_data\_preproc <- new\_data\_preproc %>%  
 mutate(issue\_d = dmy(paste0('01-', issue\_d))) %>%  
 mutate(earliest\_cr\_line = dmy(paste0('01-', earliest\_cr\_line))) %>%  
 mutate(last\_pymnt\_d = dmy(paste0('01-', last\_pymnt\_d)))  
  
## Extract the float number from int\_rate and revol\_util  
data\_preproc$int\_rate <- as.numeric(gsub("(.+)%$", "\\1", data\_preproc$int\_rate))  
data\_preproc$revol\_util <- as.numeric(gsub("(.+)%$", "\\1", data\_preproc$revol\_util))  
  
new\_data\_preproc$int\_rate <- as.numeric(gsub("(.+)%$", "\\1", new\_data\_preproc$int\_rate))  
new\_data\_preproc$revol\_util <- as.numeric(gsub("(.+)%$", "\\1", new\_data\_preproc$revol\_util))  
  
# Create new variables: Months-on-book (MOB), length of credit history  
data\_preproc <- data\_preproc %>%  
 mutate(Credit\_history = 12\*(year(issue\_d)-year(earliest\_cr\_line)) + (month(issue\_d)-month(earliest\_cr\_line))) %>%  
 mutate(MOB = 12\*(year(last\_pymnt\_d)-year(issue\_d)) + (month(last\_pymnt\_d)-month(issue\_d))) %>%  
 select(-earliest\_cr\_line)  
  
new\_data\_preproc <- new\_data\_preproc %>%  
 mutate(Credit\_history = 12\*(year(issue\_d)-year(earliest\_cr\_line)) + (month(issue\_d)-month(earliest\_cr\_line))) %>%  
 mutate(MOB = 12\*(year(last\_pymnt\_d)-year(issue\_d)) + (month(last\_pymnt\_d)-month(issue\_d))) %>%  
 select(-earliest\_cr\_line)  
  
# Convert string variables to numeric  
should\_be\_numeric <- c('dti\_joint', 'annual\_inc\_joint')  
  
for (col in should\_be\_numeric) {  
 data\_preproc[[col]] <- as.numeric(data\_preproc[[col]])  
 new\_data\_preproc[[col]] <- as.numeric(new\_data\_preproc[[col]])  
}  
  
#Convert some numeric variables to integers  
should\_be\_integers <- c( 'num\_sats', 'mths\_since\_last\_major\_derog', 'Credit\_history', 'MOB')  
  
for (col in should\_be\_integers) {  
 data\_preproc[[col]] <- as.integer(data\_preproc[[col]])  
 new\_data\_preproc[[col]] <- as.integer(new\_data\_preproc[[col]])  
}  
  
glimpse(data\_preproc)

## Observations: 1,960,050  
## Variables: 35  
## $ loan\_amnt <int> 5000, 2500, 2400, 10000, 3000, 500...  
## $ term <chr> "36 months", "60 months", "36 mont...  
## $ int\_rate <dbl> 10.65, 15.27, 15.96, 13.49, 12.69,...  
## $ installment <dbl> 162.87, 59.83, 84.33, 339.31, 67.7...  
## $ grade <chr> "B", "C", "C", "C", "B", "A", "C",...  
## $ sub\_grade <chr> "B2", "C4", "C5", "C1", "B5", "A4"...  
## $ purpose <chr> "credit\_card", "car", "small\_busin...  
## $ verification\_status <chr> "Verified", "Source Verified", "No...  
## $ home\_ownership <chr> "RENT", "RENT", "RENT", "RENT", "R...  
## $ emp\_title <chr> NA, "Ryder", NA, "AIR RESOURCES BO...  
## $ emp\_length <chr> "10+ years", "< 1 year", "10+ year...  
## $ zip\_code <chr> "860xx", "309xx", "606xx", "917xx"...  
## $ addr\_state <chr> "AZ", "GA", "IL", "CA", "OR", "AZ"...  
## $ annual\_inc <dbl> 24000.00, 30000.00, 12252.00, 4920...  
## $ dti <dbl> 27.65, 1.00, 8.72, 20.00, 17.94, 1...  
## $ dti\_joint <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ annual\_inc\_joint <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ open\_acc <int> 3, 3, 2, 10, 15, 9, 7, 4, 11, 2, 1...  
## $ total\_acc <int> 9, 4, 10, 37, 38, 12, 11, 4, 13, 3...  
## $ num\_sats <int> NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ pub\_rec <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...  
## $ revol\_bal <int> 13648, 1687, 2956, 5598, 27783, 79...  
## $ revol\_util <dbl> 83.70, 9.40, 98.50, 21.00, 53.90, ...  
## $ delinq\_2yrs <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...  
## $ recoveries <dbl> 0.00, 122.90, 0.00, 0.00, 0.00, 0....  
## $ mths\_since\_last\_delinq <int> NA, NA, NA, 35, 38, NA, NA, NA, NA...  
## $ mths\_since\_last\_major\_derog <int> NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ collections\_12\_mths\_ex\_med <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...  
## $ mths\_since\_last\_record <int> NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ issue\_d <date> 2011-12-01, 2011-12-01, 2011-12-0...  
## $ loan\_status <chr> "Fully Paid", "Charged Off", "Full...  
## $ total\_pymnt <dbl> 5863.155, 1014.530, 3005.667, 1223...  
## $ last\_pymnt\_d <date> 2015-01-01, 2013-04-01, 2014-06-0...  
## $ Credit\_history <int> 323, 152, 121, 190, 191, 85, 77, 5...  
## $ MOB <int> 37, 16, 30, 37, 61, 37, 53, 37, 4,...

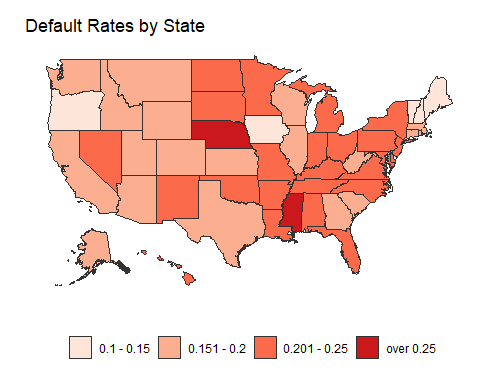
Now all the features have the right format. Before cleaning the data, we’ll investigate the number of loans and the default rate both as a function of location and time.

## Loan counts and default rates by state

data.by.states <- data\_preproc %>% filter(loan\_status %in% c('Fully Paid', 'Charged Off')) %>%  
 select(addr\_state, loan\_status) %>%  
 group\_by(addr\_state, loan\_status) %>%  
 summarise(count=n()) %>%  
 ungroup() %>%  
 spread(loan\_status, count) %>%  
 mutate(count = (`Charged Off` + `Fully Paid`), `Charged Off` = ifelse(is.na(`Charged Off`), 0, `Charged Off`), default\_ratio = `Charged Off`/ (`Charged Off` + `Fully Paid`)) %>%  
 select(count, state = addr\_state, default\_ratio, count)  
  
data.by.states$id <- tolower(state.name[match(data.by.states$state,state.abb)])  
data.by.states$id[data.by.states$state == "DC"] <- "district of columbia"  
  
# loans count  
data.by.states$count\_bins <- cut(data.by.states$count, breaks = c(0, 10000, 25000, 50000, 100000, 200000),   
 labels = c('< 10K', '10k - 25k', '25k - 50k', '50k - 100k', 'over 100k'))  
data.by.states$default\_bins <- cut(data.by.states$default\_ratio, breaks = c(0.1, 0.15, 0.2, 0.25, 1),  
 labels = c('0.1 - 0.15', '0.151 - 0.2', '0.201 - 0.25', 'over 0.25'))  
  
  
# Merge map and count/ default data  
us <- merge(fifty\_states, data.by.states, by = "id")  
  
# Create a map  
us$bucket <- cut(us$default\_ratio, breaks = c(0, 0.1, 0.15, 0.2, 1))  
  
plot\_states <- function(feature){  
 ggplot(us) +   
 geom\_map(aes(map\_id = id, fill = us[[feature]]), col = "grey20", size = .2, map = us) +   
 expand\_limits(x = fifty\_states$long, y = fifty\_states$lat) +  
 coord\_map() +  
 scale\_fill\_brewer("", palette = "Reds") +  
 scale\_x\_continuous(breaks = NULL) +   
 scale\_y\_continuous(breaks = NULL) +  
 labs(x = "", y = "") +  
 ggtitle(ifelse(grepl('default', feature), 'Default Rates by State', 'Loans by State')) +  
 theme(legend.text = element\_text(family = "Gill Sans MT"),  
 legend.position = "bottom",  
 panel.background = element\_blank(),  
 plot.title = element\_text(family = "Gill Sans MT"))  
   
}  
# grid.arrange(plot\_states('count\_bins'), plot\_states('default\_bins'), ncol = 2)  
plot\_states('count\_bins')



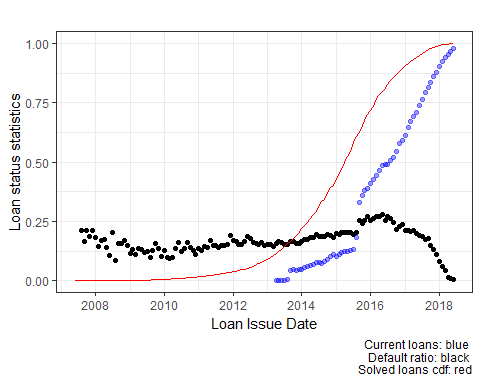
plot\_states('default\_bins')



Not surprisingly, the largest US states (CA, TX, NY, FL) have more loans than the other states. The default rate is higher in states with a few loans (Nebraska, Mississippi). It looks that the default rates are smaller in New England and the weatern states and higher for the states in-between. Later we’ll decide if use the state or geographical area for building the predictive models.

## Time-dependence of the loan status. Observations selection

vars = c('loan\_status', 'term', 'issue\_d')  
subset <- select(data\_preproc, vars)  
  
## Look at the time dependence of loan status  
loan\_status\_trend <- subset %>% select(issue\_d, loan\_status) %>%   
 group\_by(issue\_d, loan\_status) %>%  
 summarize(count = n()) %>%   
 spread(loan\_status, count) %>%   
 mutate(default\_ratio = `Charged Off`/ (`Charged Off` + `Fully Paid`), current = Current/ (Current + `Charged Off` + `Fully Paid`), sum\_solved = sum(`Charged Off`, `Fully Paid`,na.rm = TRUE)) %>%  
 ungroup() %>%  
 mutate(cum\_ratio = cumsum(sum\_solved)/sum(sum\_solved)) %>% select(-sum\_solved)  
  
# loan\_status\_trend  
  
plt1 <- ggplot(data = loan\_status\_trend, aes(x = issue\_d, y = default\_ratio), alpha = 0.4) +  
 geom\_point() +  
 geom\_point(aes(x = issue\_d, y = current), color='blue', alpha = 0.4) +  
 geom\_line(aes(x = issue\_d, y = cum\_ratio), color = 'red') +  
 theme\_bw() +  
 labs(title="", y="Loan status statistics", x="Loan Issue Date",   
 caption="Current loans: blue \n Default ratio: black \n Solved loans cdf: red")  
  
print(plt1)

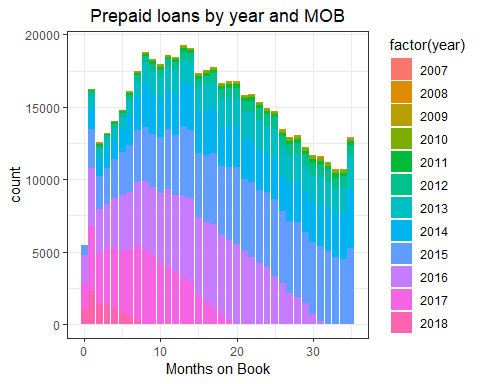


The plots in the figure above show the default rate (charged\_off/ (charged\_off + Fully\_paid)), the fraction of active loans of the total number of loans, and the cumulative curve of the loans not on the book (charged\_off or Fully\_paid), as of June 30, 2018. Each point refers to loans issued in a specific month-year. First, there seems to be a slight increase of the default rate ever time, at least for loans issued before July, 2015. What happens after that time are not *true* default rate but rather they reflect the dynamics of (re)payment. Some people default on their loans and other choose to repay early the loans, both with negative consequences on the investors’ revenue. Note that Lending Club does not charge a penalty fee for prepaying a loan.

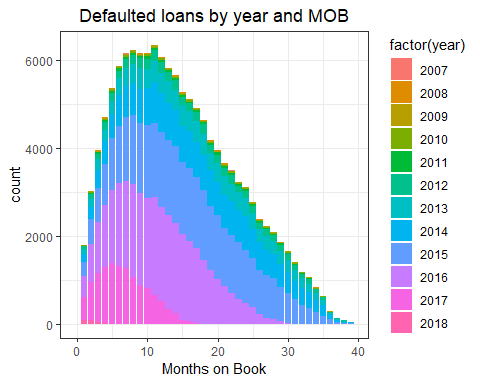
Since the underlying pattern of loan payment is different from matured loans (we have an average effect) is different from that of un-matured loans, we will use only the matured loans for the modelling phase of our study (from the red trace, they represent ~ 2/3 of data). Also, we will drop the loans issued before 2012, as they were some policy changes around that time.

Before proceeding to the next step let’s investigate the default and prepay counts as a function of time.

data\_preproc <- data\_preproc %>%  
 filter(loan\_status %in% c('Charged Off', 'Fully Paid')) %>%   
 mutate(status\_3way = ifelse(loan\_status == 'Charged Off', 'Charged.Off',  
 ifelse(((MOB < 36 & term == '36 months') |(MOB < 60 & term == '60 months')), 'Prepaid', 'Fully.Paid'))) %>%  
 mutate(loan\_status = ifelse(loan\_status == 'Fully Paid', 'Fully.Paid', 'Charged.Off'))  
  
new\_data\_preproc <- new\_data\_preproc %>%  
 filter(loan\_status %in% c('Charged Off', 'Fully Paid')) %>%   
 mutate(status\_3way = ifelse(loan\_status == 'Charged Off', 'Charged.Off',  
 ifelse(((MOB < 36 & term == '36 months') |(MOB < 60 & term == '60 months')), 'Prepaid', 'Fully.Paid'))) %>%  
 mutate(loan\_status = ifelse(loan\_status == 'Fully Paid', 'Fully.Paid', 'Charged.Off'))  
  
vars = c('loan\_status', 'status\_3way', 'term', 'issue\_d', 'grade', 'MOB')  
  
subset <- select(data\_preproc, vars) %>% filter(term == '36 months') %>%  
 mutate(year = year(issue\_d))  
  
prepaid <- subset %>% filter(status\_3way == 'Prepaid')  
  
plt2 <- ggplot(data = prepaid, aes(MOB)) +  
 geom\_bar(aes(fill=factor(year))) +  
 theme\_bw() +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 ggtitle('Prepaid loans by year and MOB') + xlab('Months on Book')  
  
print(plt2)



## Investigate the defaulted loans  
  
defaulted <- subset %>% filter(status\_3way == 'Charged.Off')   
  
plt3 <- ggplot(data = defaulted[defaulted$MOB <40,], aes(MOB)) +  
 geom\_bar(aes(fill=factor(year))) +  
 theme\_bw() +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 ggtitle('Defaulted loans by year and MOB') + xlab('Months on Book')  
  
print(plt3)



On average, a borrower default on a 36-month loan after 14.5 or prepays the loan in 16.8 months.

We also analysed the default and the prepay distribution as a function of MOB and loan grade or the amount borrowed and did not find significant differences between payment patterns.

Next, we’ll reduce the number of observations to include only the matured loans issued after January 1, 2012. This will account for ~ 60% of the loans that were either fully paid or charged off. We chose the January 2012 as our lower limit because some changes were made in Lending Club policy around that time. Technically, there should not be any current loans left.

data\_short <- data\_preproc %>% filter(issue\_d >= ymd('2012-01-01')) %>%  
 filter((term == '36 months' & issue\_d < ymd('2015-07-01')) | (term == '60 months' & issue\_d < ymd('2013-07-01'))) %>%  
 select(-last\_pymnt\_d, -MOB)  
  
new\_data\_short <- new\_data\_preproc %>% filter(issue\_d >= ymd('2015-07-01') & issue\_d <= ymd('2015-09-01')) %>%  
 select(-last\_pymnt\_d, -MOB)  
  
knitr::kable(table(data\_short$term), caption = 'Table2: Frequencies of loan durations')

Table2: Frequencies of loan durations

|  |  |
| --- | --- |
| Var1 | Freq |
| 36 months | 427079 |
| 60 months | 22137 |

table(data\_short$loan\_status)

##   
## Charged.Off Fully.Paid   
## 64688 384528

We have less than 5% observations on 60-months loans, we’ll continue only with the 36-months loans.

data\_short <- data\_short %>% filter(term == '36 months') %>%  
 select(-term)  
new\_data <- new\_data\_short %>% filter(term == '36 months') %>%  
 select(-term)

At this point we redefine our research question: **Design a classification algorithm to predict defaulting loans for short period loans offered by Lending Club**

# 2. Data Pre-processing

## Additional feature selection

In this step we’ll remove features based on the following criteria: (a) more than 50% missing values; (b) categorical features with more than 51 levels, and (c) zero- and near-zero variance (95%)

1. missing values

col\_nas = apply(is.na(data\_short), 2, sum)  
cols\_with\_nas <- colnames(data\_short)[col\_nas/nrow(data\_short) > 0.5]  
  
data\_short <- data\_short %>% select(-cols\_with\_nas)  
new\_data <- new\_data %>% select(-cols\_with\_nas)

Removed 5 feature(s).

1. too many levels

n\_levels <- colnames(data\_short)[sapply(data\_short, function(x) {is.character(x) & (length(unique(x)) > 60)})]  
data\_short <- data\_short %>% select(-n\_levels)  
new\_data <- new\_data %>% select(-n\_levels)

Removed 2 feature(s).

1. low variance data

low\_var <- nearZeroVar(data\_short, freqCut = 95/5, saveMetrics = T)  
cols\_with\_nzv <- colnames(data\_short)[low\_var$nzv]  
data\_short <- data\_short %>% select(-cols\_with\_nzv)  
new\_data <- new\_data %>% select(-cols\_with\_nzv)

Dropped 2 feature(s).

Data summary

glimpse(data\_short)

## Observations: 427,079  
## Variables: 24  
## $ loan\_amnt <int> 7550, 8000, 27050, 4800, 12000, 12000, 280...  
## $ int\_rate <dbl> 16.24, 10.99, 10.99, 10.99, 7.62, 11.99, 7...  
## $ installment <dbl> 266.34, 261.88, 885.46, 157.13, 373.94, 39...  
## $ grade <chr> "C", "B", "B", "B", "A", "B", "A", "B", "B...  
## $ sub\_grade <chr> "C5", "B2", "B2", "B2", "A3", "B3", "A3", ...  
## $ purpose <chr> "debt\_consolidation", "debt\_consolidation"...  
## $ verification\_status <chr> "Not Verified", "Not Verified", "Verified"...  
## $ home\_ownership <chr> "RENT", "MORTGAGE", "OWN", "MORTGAGE", "MO...  
## $ emp\_length <chr> "3 years", "2 years", "10+ years", "2 year...  
## $ addr\_state <chr> "CA", "CO", "MI", "TX", "TX", "CO", "CA", ...  
## $ annual\_inc <dbl> 28000, 33000, 55000, 39600, 96500, 130000,...  
## $ dti <dbl> 8.40, 15.75, 22.87, 2.49, 12.61, 13.03, 18...  
## $ open\_acc <int> 4, 9, 14, 3, 17, 9, 15, 14, 7, 6, 8, 12, 5...  
## $ total\_acc <int> 5, 16, 27, 8, 30, 19, 31, 39, 32, 14, 29, ...  
## $ num\_sats <int> 4, 9, 14, 3, 17, 9, 15, 14, 7, 6, 8, 12, 5...  
## $ pub\_rec <int> 0, 1, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 2, 0, ...  
## $ revol\_bal <int> 5759, 7203, 36638, 4136, 13248, 10805, 295...  
## $ revol\_util <dbl> 72.0, 34.6, 61.2, 16.1, 55.7, 67.0, 54.6, ...  
## $ delinq\_2yrs <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 2, ...  
## $ issue\_d <date> 2013-12-01, 2013-12-01, 2013-12-01, 2013-...  
## $ loan\_status <chr> "Fully.Paid", "Charged.Off", "Fully.Paid",...  
## $ total\_pymnt <dbl> 9600.455, 5622.500, 31752.530, 5157.519, 1...  
## $ Credit\_history <int> 38, 269, 326, 220, 123, 193, 229, 299, 182...  
## $ status\_3way <chr> "Fully.Paid", "Charged.Off", "Prepaid", "P...

We have now a dataset with 427079 observation, 21 predictive features, and 3 (potential) target variables: *loan\_status*, *status\_3way*, *total\_pymnt*.

## Add features with socio-economic data

In this step we create new features that are related to the time the loan was issued or the location (state). This features are:

* unemployment rate (<https://download.bls.gov/pub/time.series/ln/ln.data.1.AllData>)
* credit card rates (alternative way to obtain credit)
* S&P 500 (indicate the economy’s health) (<https://ca.finance.yahoo.com/quote/%5EGSPC/history?p=%5EGSPC>)
* Treasury Bills (estimate future trends in economy) (<https://www.treasury.gov/resource-center/data-chart-center/interest-rates/Pages/TextView.aspx?data=yield>)

unemp <-read\_csv('C:\\Ryerson Capstone\\data\\unemployment\_data\_clean.csv')  
tbills <-read\_csv('C:\\Ryerson Capstone\\data\\tbills\_clean.csv')  
sp500 <- read\_csv('C:\\Ryerson Capstone\\data\\SP500\_clean.csv')  
creditCards <- read\_csv('C:\\Ryerson Capstone\\data\\Credit card avg interest rates.csv')  
  
data\_short <- data\_short %>% mutate(Date = issue\_d, State = addr\_state) %>%   
 inner\_join(sp500, by = 'Date') %>%  
 inner\_join(tbills, by = 'Date') %>%  
 inner\_join(creditCards, by = 'Date') %>%  
 inner\_join(unemp, by = c('State', 'Date')) %>%  
 select(-Date, -State, -issue\_d)  
  
new\_data <- new\_data %>% mutate(Date = issue\_d, State = addr\_state) %>%   
 inner\_join(sp500, by = 'Date') %>%  
 inner\_join(tbills, by = 'Date') %>%  
 inner\_join(creditCards, by = 'Date') %>%  
 inner\_join(unemp, by = c('State', 'Date')) %>%  
 select(-Date, -State, -issue\_d)

Some of the added features are strongly correlated. We’ll deal with the correlations between the numerical variables at a later stage.

## Group states by region

states <-read\_csv('C:\\Ryerson Capstone\\data\\state\_codes.csv')  
data\_short <- data\_short %>% mutate(State\_abv = addr\_state) %>%   
 inner\_join(states, by = 'State\_abv') %>%  
 select(-addr\_state, -State\_code, -State) %>%  
 select(everything(), State = State\_abv)  
  
new\_data <- new\_data %>% mutate(State\_abv = addr\_state) %>%   
 inner\_join(states, by = 'State\_abv') %>%  
 select(-addr\_state, -State\_code, -State) %>%  
 select(everything(), State = State\_abv)

## Split the data into train - validation - test sets

## SPLIT THE DATA INTO TRAIN/ VALIDATION/ TEST SETS. We'll deal with the missing values later  
  
set.seed(136)  
  
split <- sample.split(data\_short$loan\_status, SplitRatio = .6)  
train <- data\_short[split==TRUE, ] %>% mutate(set = 'train')  
vt <- data\_short[split==FALSE, ]  
  
set.seed(136)  
  
split <- sample.split(vt$loan\_status, SplitRatio = .5)  
validation <- vt[split==TRUE,] %>% mutate(set = 'validation')  
test <- vt[split==FALSE,] %>% mutate(set = 'test')  
  
data\_split <- rbind(train, validation, test)  
  
write\_csv(data\_split, 'C:\\Ryerson Capstone\\data\\LC\\data\_split.csv')  
write\_csv(new\_data, 'C:\\Ryerson Capstone\\data\\LC\\new\_data.csv')

For simplicity, we rename our dataset: data - combined train-validation-test dataset and new\_data (data after June 2015).

data <- data\_split %>% select(-total\_pymnt) %>% select(loan\_status, everything()) %>%  
 select(everything(), status\_3way)  
new\_data <- new\_data%>% select(-total\_pymnt) %>% select(loan\_status, everything()) %>%  
 select(everything(), status\_3way)  
  
train <- data[data$set == 'train',] %>% select(-set)  
# summary(train)  
  
# clean the work space  
# rm(list = setdiff(ls(), c('data', 'new\_data', 'train')))

## Missing data imputation

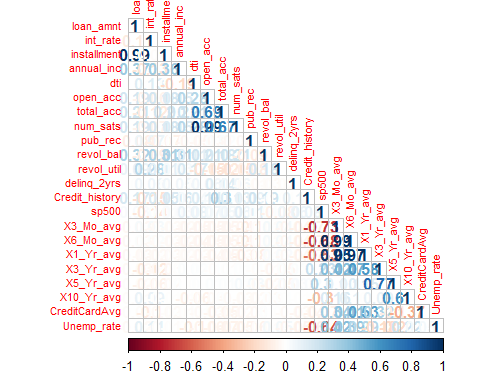
imputer <- function(x) {  
 if (is.character(x)) return (names(which.max(table(x))))  
 else return(round(median(x, na.rm = TRUE), 2))  
}  
  
impute\_values <- lapply(train[,1:ncol(train)], imputer)  
  
# save values for feature needs  
save(impute\_values, file = 'C:\\Ryerson Capstone\\data\\LC\\impute\_values.RData')  
  
for (i in 1:(ncol(data)-1)) {  
 data[is.na(data[[i]]), i] <- impute\_values[[i]]  
 new\_data[is.na(new\_data[[i]]), i] <- impute\_values[[i]]  
}  
  
train <- data[data$set == 'train',] %>% select(-set)  
paste('Are NAs in the data? ', anyNA(data))

## [1] "Are NAs in the data? FALSE"

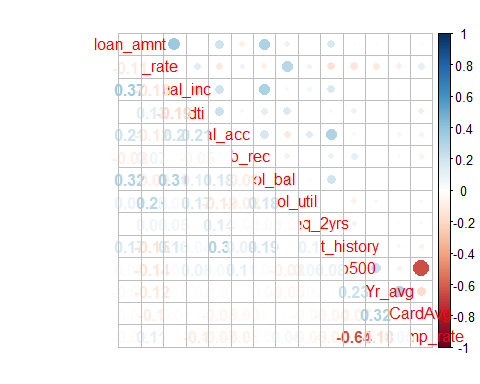
## Analysis of the numeric independent in the data

### Remove numeric variables that are highly correlated

library(corrplot)  
num.cols <- sapply(train, function(x) {is.numeric(x) | is.integer(x)})  
  
pearsoncor <- cor(train[num.cols], use="complete.obs")  
corrplot(pearsoncor, method = "number", type = 'lower', tl.cex = 0.7, cl.cex = 0.8)



#remove features highly correlated (> 0.8)  
features <- c('installment', 'num\_sats', 'X3\_Mo\_avg', 'X6\_Mo\_avg', 'X5\_Yr\_avg', 'X1\_Yr\_avg', 'X10\_Yr\_avg', 'open\_acc')  
pearsoncor <- cor(select(train[num.cols], -features), use="complete.obs")  
corrplot(pearsoncor, type = 'upper',tl.pos = 'd')  
corrplot(pearsoncor, add = TRUE, type = 'lower', method = "number", diag = FALSE, tl.pos = 'n', cl.pos = 'n', tl.cex = 0.5)



data <- select(data, -features)  
new\_data <- select(new\_data, -features)  
train <- data[data$set == 'train',] %>% select(-set)

Helper functions

labels = c("Loan status", "Loan amount", "Interest rate", "Loan grade", "Loan subgrade", "Purpose",  
 "Verification status", "Home ownership", "Employment length", "Annual income", "Debt-to-Income ratio",  
 "Total accounts", "Public records", "Balance revolving accounts", "% Revolving accounts limit used",  
 "Delinquencies last 2 years", "Months of credit history", "S&P500 monthly average ",   
 "Average rate 3-month T-bills", "Average credit card rate", "Unemployment rate", 'State', 'Area')  
  
plot.numeric <- function(index, df = train) {  
  
 plt1 <- ggplot(data = df) +  
 geom\_histogram(aes(df[[index]], fill=loan\_status)) +  
 theme\_bw() +  
 scale\_x\_continuous() +  
 xlab(labels[index])  
 plt2 <- ggplot(data = df) +  
 geom\_boxplot(aes(loan\_status, df[[index]])) +  
 coord\_flip() +  
 theme\_bw() +  
 scale\_y\_continuous() +  
 scale\_x\_discrete('Loan Status') +  
 ylab(labels[index])  
 grid.arrange(plt1, plt2, ncol = 1)  
}  
  
transform <- function(vec) {  
 # ref = abs(skewness(vec))  
 Transformation = c('no change', 'square root', 'natural log', 'inverse square root')  
 mat = sapply((vec + .01), function(x) {c(no\_change = x, square\_root = sqrt(x), log\_n = log(x), inv\_sq\_root = 1/sqrt(x))})  
 skew = apply(mat, 1, skewness)  
 print(data.frame(Skew = round(skew, 3)))  
}  
  
data.transformed <- data %>% select(one\_of(c('loan\_status', 'set')))  
new\_data.transformed <- new\_data %>% select(loan\_status)

## Numeric data summary

num.cols <- sapply(train, function(x) {is.numeric(x) | is.integer(x)})  
numeric.vars <- train[num.cols]  
default <- train$loan\_status == 'Charged.Off'  
  
data.normal <- function(x){  
 if (ad.test(x)$p.value > 0.05){  
 return('TRUE')  
 } else {  
 return('FALSE')  
 }  
}  
wcx <- function(x) {  
 if(wilcox.test(x[default],x[!default])$p.value <= 0.05) {  
 return('TRUE')  
 } else {  
 return('FALSE')  
 }  
}  
  
mins <- round(sapply(numeric.vars, min),1)  
maxs <- round(sapply(numeric.vars, max),1)  
skew <- round(sapply(numeric.vars, skewness),1)  
means <- round(sapply(numeric.vars, mean), 1)  
st.devs <- round(sapply(numeric.vars, sd), 1)  
medians <- round(sapply(numeric.vars, median), 1)  
normality <- sapply(numeric.vars, data.normal)  
Wilcox <- sapply(numeric.vars, wcx)  
  
numeric.summary <- data.frame(Min.Val = mins, Mean = means, Median = medians, Max.Val = maxs,  
 StDev = st.devs, Skew = skew, Wilcox = Wilcox)  
  
knitr::kable(numeric.summary)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Min.Val | Mean | Median | Max.Val | StDev | Skew | Wilcox |
| loan\_amnt | 1000.0 | 12557.4 | 10000.0 | 35000.0 | 7674.6 | 1.0 | TRUE |
| int\_rate | 5.3 | 12.4 | 12.3 | 27.9 | 3.9 | 0.4 | TRUE |
| annual\_inc | 3000.0 | 71716.3 | 60000.0 | 7500000.0 | 59917.5 | 38.3 | TRUE |
| dti | 0.0 | 17.5 | 17.0 | 40.0 | 8.1 | 0.2 | TRUE |
| total\_acc | 2.0 | 24.7 | 23.0 | 156.0 | 11.7 | 0.9 | TRUE |
| pub\_rec | 0.0 | 0.2 | 0.0 | 54.0 | 0.6 | 12.0 | TRUE |
| revol\_bal | 0.0 | 15411.0 | 10767.0 | 2904836.0 | 22331.9 | 30.5 | TRUE |
| revol\_util | 0.0 | 55.0 | 55.8 | 892.3 | 23.4 | 0.0 | TRUE |
| delinq\_2yrs | 0.0 | 0.3 | 0.0 | 29.0 | 0.9 | 5.5 | TRUE |
| Credit\_history | 36.0 | 192.1 | 174.0 | 793.0 | 90.2 | 1.1 | TRUE |
| sp500 | 1300.6 | 1856.0 | 1937.3 | 2111.9 | 227.6 | -0.8 | TRUE |
| X3\_Yr\_avg | 0.3 | 0.9 | 0.9 | 1.3 | 0.2 | -0.6 | TRUE |
| CreditCardAvg | 11.8 | 11.9 | 11.9 | 12.4 | 0.1 | 1.5 | TRUE |
| Unemp\_rate | 2.9 | 6.6 | 6.4 | 12.3 | 1.4 | 0.4 | TRUE |

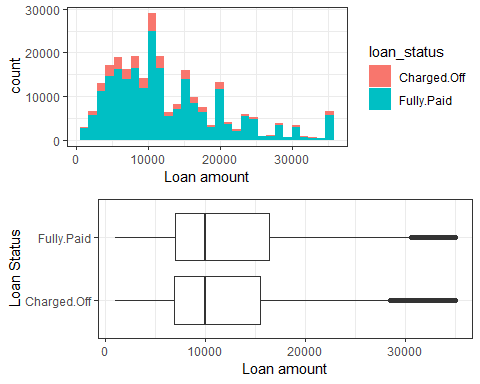
The table above shows the five-number statistics of the numeric variables, as well as the skewness, and the results of the Wilcox test on independenceconditioned on the dependent variable. Data normality test (Anderson - Darling), shows that none of the numerical variables has a normal distribution.

In the following section we’ll look at the numeric data distributions conditioned on the *loan\_status* dependent variable and apply numerical transformations based on a simplified Box-Cox (like) approach, i.e. find the lowest skewness among raw data (x), logarithm -, square root -, and inverse square root - transformed data.

## Numerical feature description

### 1. Loan amount (loan\_amnt)

plot.numeric(grep('loan\_amnt', colnames(train)))



transform(train$loan\_amnt)

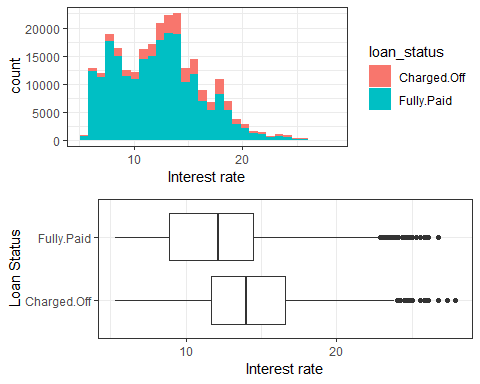
## Skew  
## no\_change 1.040  
## square\_root 0.364  
## log\_n -0.507  
## inv\_sq\_root 1.796

Transformation: square root

data.transformed$loan\_amnt <- scale(sqrt(data$loan\_amnt))  
new\_data.transformed$loan\_amnt <- scale(sqrt(new\_data$loan\_amnt))

### 2. Interest rate (int\_rate)

plot.numeric(grep('int\_rate', colnames(train)))



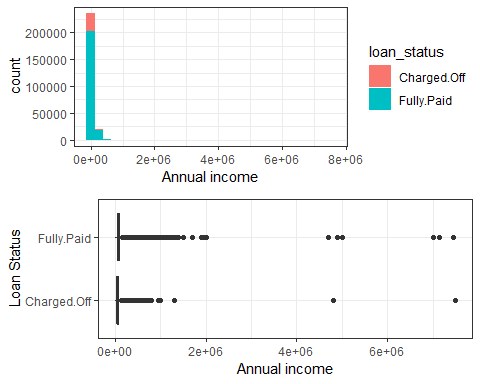
# transform(train$int\_rate)

Transformation: square root

data.transformed$int\_rate <- scale(sqrt(data$int\_rate))  
new\_data.transformed$int\_rate <- scale(sqrt(new\_data$int\_rate))

### 3. Annual income (annual\_inc)

plot.numeric(grep('annual\_inc', colnames(train)))



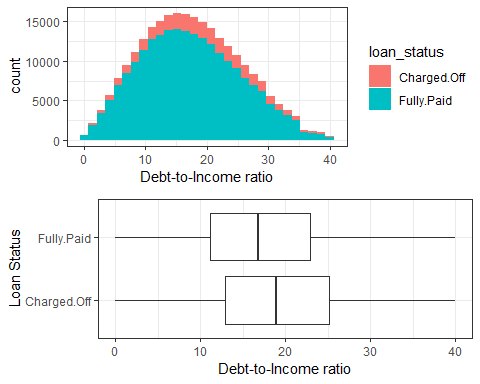
# transform(train$annual\_inc)

Transformation: log There are 256 borrowers with income above half a million. Will clip the annual income to 500000.

data$annual\_inc <- ifelse(data$annual\_inc > 500000, 500000, data$annual\_inc)  
new\_data$annual\_inc <- ifelse(new\_data$annual\_inc > 500000, 500000, new\_data$annual\_inc)  
data.transformed$annual\_inc <- scale(log(data$annual\_inc + 1))  
new\_data.transformed$annual\_inc <- scale(log(new\_data$annual\_inc+1))

### 4. Debt-to-Income ratio (dti)

plot.numeric(grep('dti', colnames(train)))



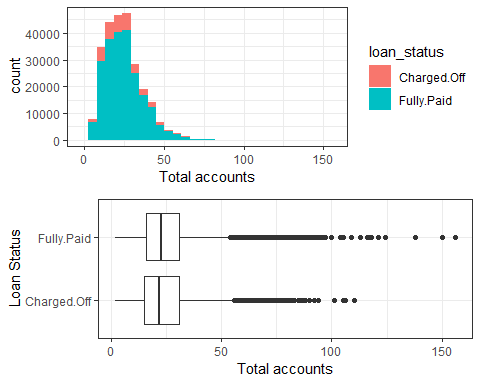
# transform(train$dti)

Transformation: None

data.transformed$dti <- scale(data$dti)  
new\_data.transformed$dti <- scale(new\_data$dti)

### 5. Total accounts (total\_acc)

plot.numeric(grep('total\_acc', colnames(train)))



transform(train$total\_acc)

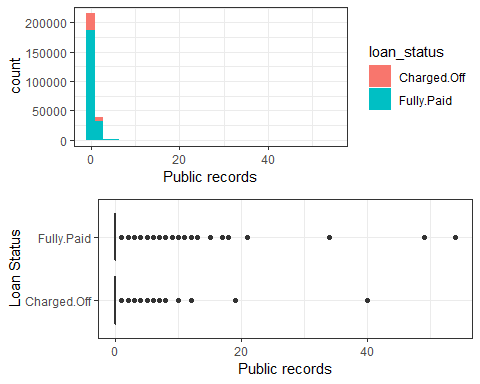
## Skew  
## no\_change 0.889  
## square\_root 0.209  
## log\_n -0.497  
## inv\_sq\_root 1.361

Transformation: square root

data.transformed$total\_acc <- scale(sqrt(data$total\_acc))  
new\_data.transformed$total\_acc <- scale(sqrt(new\_data$total\_acc))

### 6. Public records (pub\_rec)

plot.numeric(grep('pub\_rec', colnames(train)))



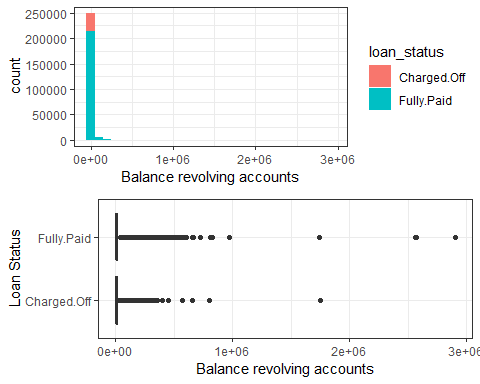
# transform(train$pub\_rec)

Transformation: square root

data.transformed$pub\_rec <- scale(sqrt(data$pub\_rec))  
new\_data.transformed$pub\_rec <- scale(sqrt(new\_data$pub\_rec))

### 7. Revolving balance (revol\_bal)

plot.numeric(grep('revol\_bal', colnames(train)))



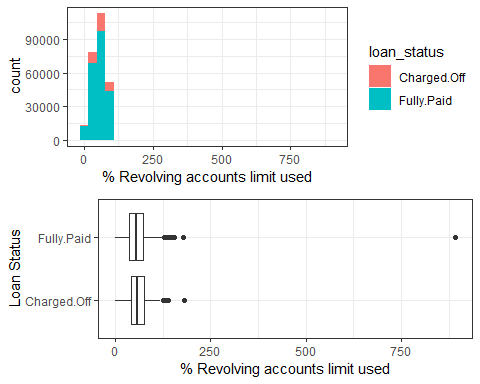
# transform(train$revol\_bal)

Transformation: (lambda = 0.25)

data.transformed$revol\_bal <- scale(data$revol\_bal^(1/4))  
new\_data.transformed$revol\_bal <- scale(new\_data$revol\_bal^(1/4))

### 8. Revolving utilization ratio (revol\_util)

plot.numeric(grep('revol\_util', colnames(train)))



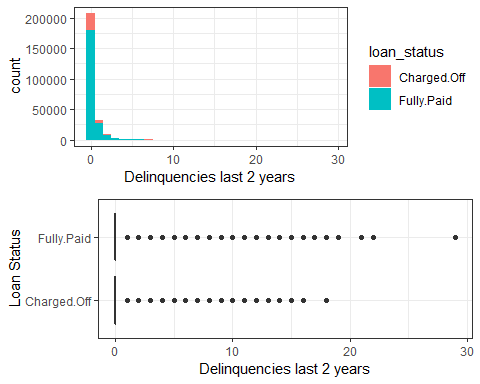
# transform(train$revol\_util)

Transformation: None

data.transformed$revol\_util <- scale(data$revol\_util)  
new\_data.transformed$revol\_util <- scale(new\_data$revol\_util)

### 9. Delinquencies in the last two years (delinq\_2yrs)

plot.numeric(grep('delinq\_2yrs', colnames(train)))



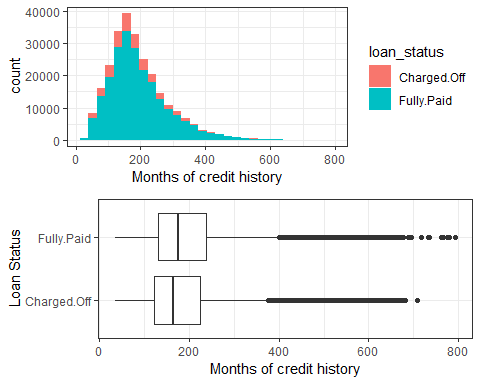
# transform(train$delinq\_2yrs)

Transformation: log

data.transformed$delinq\_2yrs <- scale(log(1+data$delinq\_2yrs))  
new\_data.transformed$delinq\_2yrs <- scale(log(1+new\_data$delinq\_2yrs))

### 10. Lentgh of credit history (Credit\_history)

plot.numeric(grep('Credit\_history', colnames(train)))



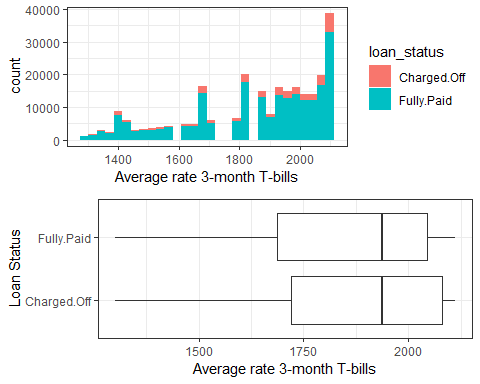
# transform(train$Credit\_history)

Transformation: log

data.transformed$Credit\_history <- scale(log(data$Credit\_history))  
new\_data.transformed$Credit\_history <- scale(log(new\_data$Credit\_history))

### 11. S&P500 monthly average (sp500)

plot.numeric(grep('sp500', colnames(train)))



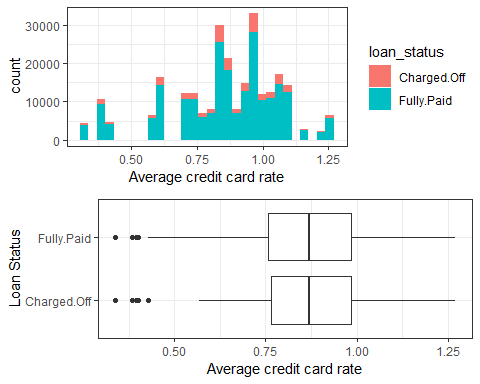
# transform(train$sp500)

Transformation: None

data.transformed$sp500 <- scale(data$sp500)  
new\_data.transformed$sp500 <- scale(new\_data$sp500)

### 12. Treasury Bills 3 years interest (X3\_Yr\_avg)

plot.numeric(grep('X3\_Yr\_avg', colnames(train)))



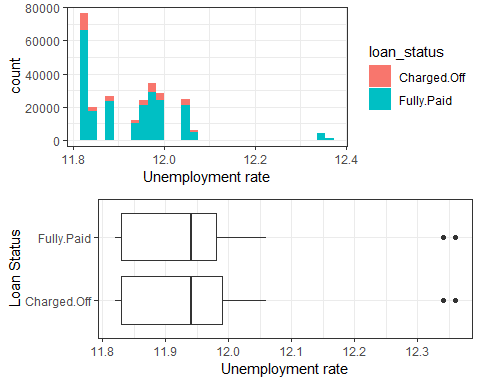
# transform(train$X3\_Yr\_avg)

Transformation: None

data.transformed$X3\_Yr\_avg <- scale(data$X3\_Yr\_avg)  
new\_data.transformed$X3\_Yr\_avg <- scale(new\_data$X3\_Yr\_avg)

### 13. Monthly credit card interest rate (CreditCardAvg)

plot.numeric(grep('CreditCardAvg', colnames(train)))



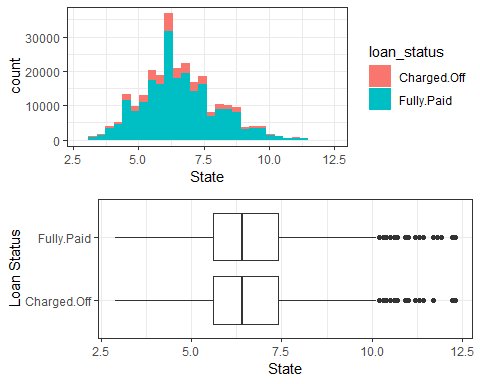
# transform(train$CreditCardAvg)

Transformation: None

data.transformed$CreditCardAvg <- scale(data$CreditCardAvg)  
new\_data.transformed$CreditCardAvg <- scale(new\_data$CreditCardAvg)

### 14. Unemployment rate, by state and month-year (Unemp\_rate)

plot.numeric(grep('Unemp\_rate', colnames(train)))



# transform(train$Unemp\_rate)

Transformation: square root

data.transformed$Unemp\_rate <- scale(sqrt(data$Unemp\_rate))  
new\_data.transformed$Unemp\_rate <- scale(sqrt(new\_data$Unemp\_rate))

## Analysis of the categorical (factor) independent variables

### Variable independence

factor.cols <- sapply(train, function(x) {is.character(x)})  
  
chisqmatrix <- function(x) {  
 names = colnames(x); num = length(names)  
 m = matrix(nrow=num,ncol=num,dimnames=list(names,names))  
 for (i in 1:(num-1)) {  
 for (j in (i+1):num) {  
 m[i,j] = chisq.test(x[[i]],x[[j]])$p.value  
 }  
 }  
 return (m)  
}  
  
  
x <- train[,factor.cols]  
mat = chisqmatrix(x)  
print(mat)

## loan\_status grade sub\_grade purpose  
## loan\_status NA 0 0 7.176016e-139  
## grade NA NA 0 0.000000e+00  
## sub\_grade NA NA NA 0.000000e+00  
## purpose NA NA NA NA  
## verification\_status NA NA NA NA  
## home\_ownership NA NA NA NA  
## emp\_length NA NA NA NA  
## status\_3way NA NA NA NA  
## State NA NA NA NA  
## Area NA NA NA NA  
## verification\_status home\_ownership emp\_length  
## loan\_status 4.42358e-98 1.000523e-254 1.162495e-120  
## grade 0.00000e+00 0.000000e+00 3.434933e-119  
## sub\_grade 0.00000e+00 0.000000e+00 2.663918e-79  
## purpose 0.00000e+00 0.000000e+00 1.938754e-277  
## verification\_status NA 4.423298e-257 0.000000e+00  
## home\_ownership NA NA 0.000000e+00  
## emp\_length NA NA NA  
## status\_3way NA NA NA  
## State NA NA NA  
## Area NA NA NA  
## status\_3way State Area  
## loan\_status 0.000000e+00 8.132385e-66 7.770211e-16  
## grade 0.000000e+00 3.371944e-77 1.406892e-45  
## sub\_grade 0.000000e+00 4.413406e-39 1.806702e-27  
## purpose 8.486003e-234 5.232540e-219 2.705436e-193  
## verification\_status 1.321632e-101 3.907482e-47 3.963081e-24  
## home\_ownership 2.079710e-282 0.000000e+00 0.000000e+00  
## emp\_length 3.096111e-262 3.561945e-260 7.578080e-143  
## status\_3way NA 2.787073e-175 1.197235e-110  
## State NA NA 0.000000e+00  
## Area NA NA NA

It looks that all the categorical variables are independent.

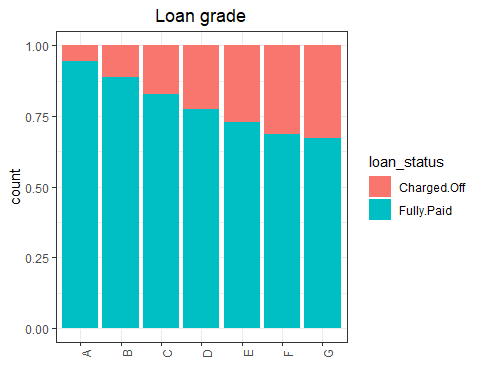
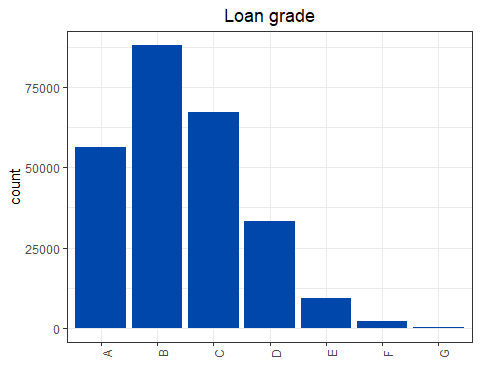
Helper functions

factor.vars <- train[!num.cols]  
  
plot.factor <- function(index, df = train) {  
 plt1 <- ggplot(data = df) +  
 geom\_bar(aes(df[[index]]), fill='#0047AB') +  
 theme\_bw() +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 theme(axis.text.x = element\_text(angle=90, hjust = 0.5)) +  
 ggtitle(labels[index]) + xlab('')  
   
 print(plt1)  
   
 plt2 <- ggplot(data = df) +  
 geom\_bar(aes(df[[index]], fill=loan\_status), position = 'fill') +  
 theme\_bw() +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 theme(axis.text.x = element\_text(angle=90, hjust = 0.5)) +  
 ggtitle(labels[index]) + xlab('')  
   
 print(plt2)  
 # grid.arrange(plt1, plt2, ncol = 1)  
}

## Categorical (factor) variables

### 1. Loan grade (grade)

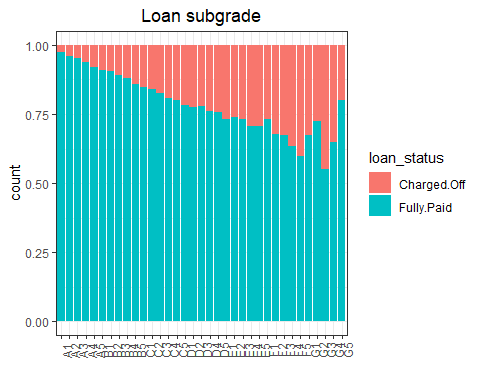
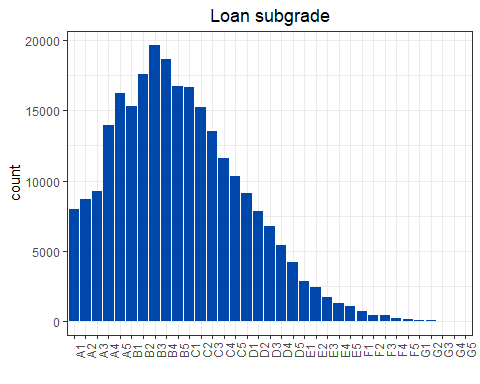
plot.factor(grep('^[g]rade', colnames(train)))



data.transformed$grade <- data$grade  
new\_data.transformed$grade <- new\_data$grade

### 2. Loan sub-grade (sub\_grade)

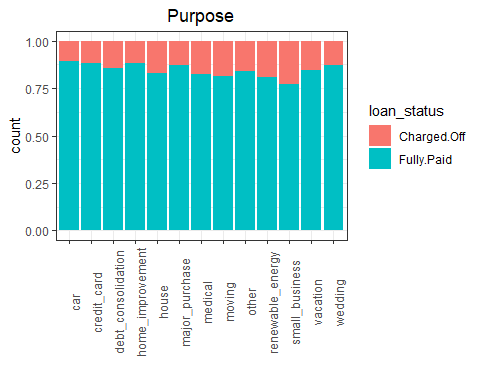
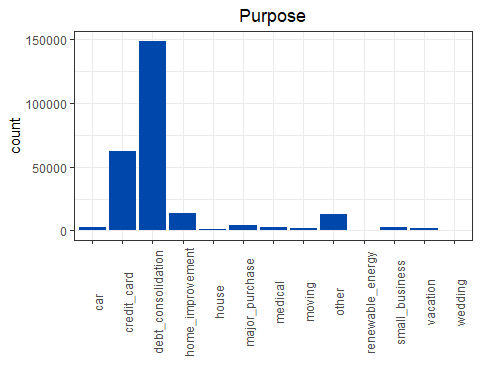
plot.factor(grep('sub\_grade', colnames(train)))



data.transformed$sub\_grade <- data$sub\_grade  
new\_data.transformed$sub\_grade <- new\_data$sub\_grade

### 3. Loan purpose (purpose)

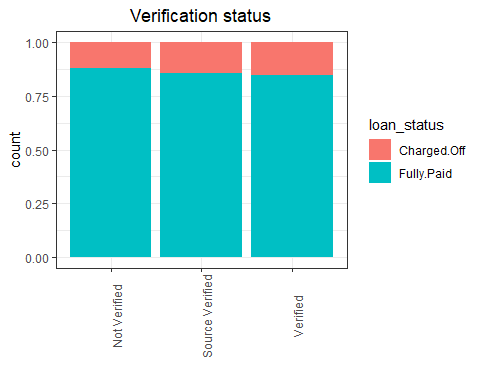
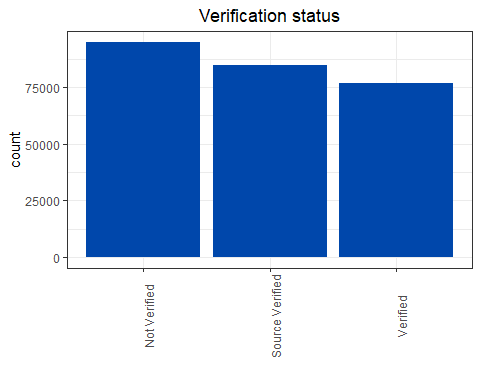
plot.factor(grep('purpose', colnames(train)))



data.transformed$purpose <- data$purpose  
new\_data.transformed$purpose <- new\_data$purpose

### 4. Verification status (verification\_status)

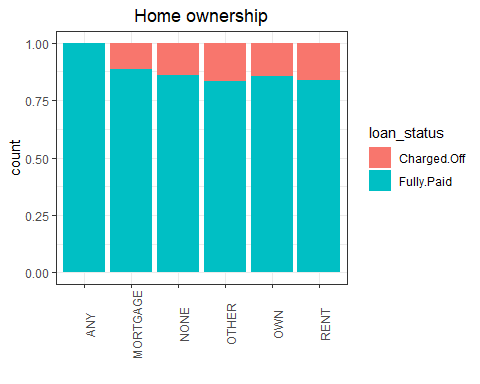
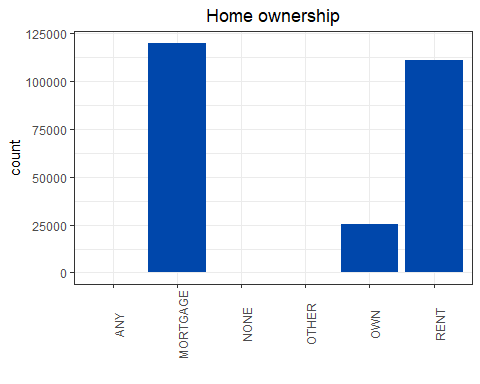
plot.factor(grep('verification\_status', colnames(train)))



data.transformed$verification\_status <- data$verification\_status  
new\_data.transformed$verification\_status <- new\_data$verification\_status

### 5. Home ownership (home\_ownership)

plot.factor(grep('home\_ownership', colnames(train)))

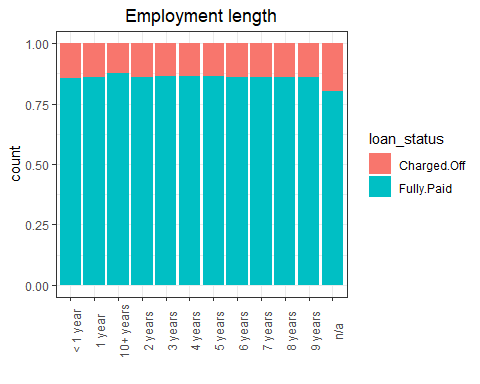
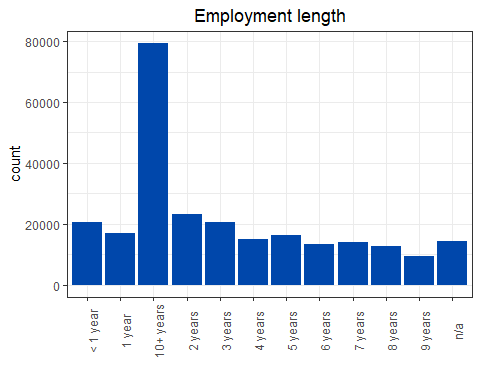


Consolidate the minor groups into ‘OTHER’

data$home\_ownership <- ifelse(data$home\_ownership %in% c('ANY', 'NONE', 'OTHER'), 'OTHER', data$home\_ownership)  
new\_data$home\_ownership <- ifelse(new\_data$home\_ownership %in% c('ANY', 'NONE', 'OTHER'), 'OTHER', new\_data$home\_ownership)  
data.transformed$home\_ownership <- ifelse(data$home\_ownership %in% c('ANY', 'NONE', 'OTHER'), 'OTHER', data$home\_ownership)  
new\_data.transformed$home\_ownership <- ifelse(new\_data$home\_ownership %in% c('ANY', 'NONE', 'OTHER'), 'OTHER', new\_data$home\_ownership)

### 6. Employment length (emp\_length)

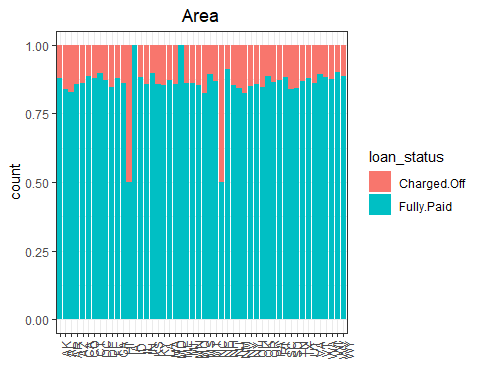
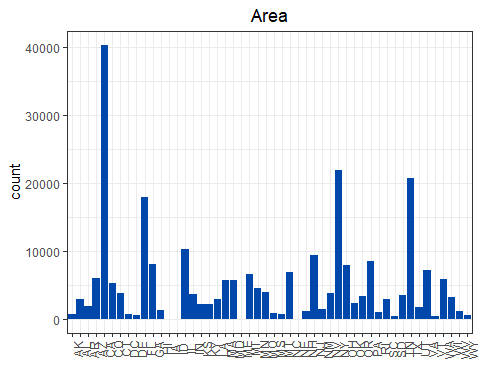
plot.factor(grep('emp\_length', colnames(train)))

 Gather the employment length in fewer levels, as below.

data$emp\_length <- ifelse(data$emp\_length %in% c("< 1 year", "1 year", "2 years"), "up to 2yr",   
 ifelse(data$emp\_length %in% c("3 years", "4 years", "5 years"), "3-5 yrs",   
 ifelse(data$emp\_length %in% c("6 years", "7 years", "8 years", "9 years"), "6-9 yrs",  
 data$emp\_length)))  
new\_data$emp\_length <- ifelse(new\_data$emp\_length %in% c("< 1 year", "1 year", "2 years"), "up to 2yr",   
 ifelse(new\_data$emp\_length %in% c("3 years", "4 years", "5 years"), "3-5 yrs",   
 ifelse(new\_data$emp\_length %in% c("6 years", "7 years", "8 years", "9 years"), "6-9 yrs",  
 new\_data$emp\_length)))  
data.transformed$emp\_length <- data$emp\_length  
new\_data.transformed$emp\_length <- new\_data$emp\_length

### 7. State (State)

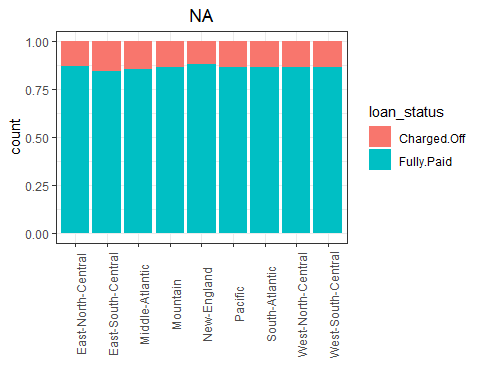
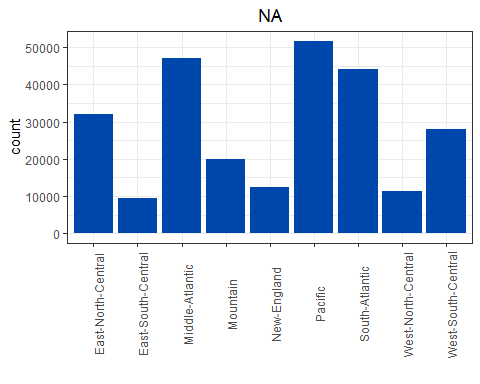
plot.factor(grep('State', colnames(train)))



data.transformed$State <- data$State  
new\_data.transformed$State <- new\_data$State

### 8. Geographical area (Area)

plot.factor(grep('Area', colnames(train)))



data.transformed$Area <- data$Area  
new\_data.transformed$Area <- new\_data$Area

# Save the final files for the modelling phase

data.transformed$status\_3way <- data$status\_3way  
new\_data.transformed$status\_3way <- new\_data$status\_3way  
write\_csv(data, 'LC\_data\_untransformed.csv')  
write\_csv(new\_data, 'LC\_new\_data\_untransformed.csv')  
write\_csv(data.transformed, 'LC\_data\_transformed.csv')  
write\_csv(new\_data.transformed, 'LC\_new\_data\_transformed.csv')

# Further feature selection using the Boruta package

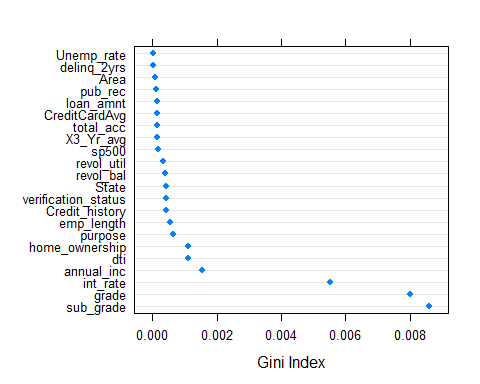
library(Boruta)  
set.seed(101)  
  
if(file.exists('C:\\Ryerson Capstone\\data\\LC\\boruta.RData')) {  
 load('C:\\Ryerson Capstone\\data\\LC\\boruta.RData')  
} else {  
 boruta.train <- Boruta(factor(loan\_status) ~., data = train[,1:23], doTrace = 2)  
}  
  
print(boruta.train)

## Boruta performed 12 iterations in 1.946476 hours.  
## 21 attributes confirmed important: addr\_state, annual\_inc,  
## Credit\_history, CreditCardAvg, delinq\_2yrs and 16 more;  
## No attributes deemed unimportant.

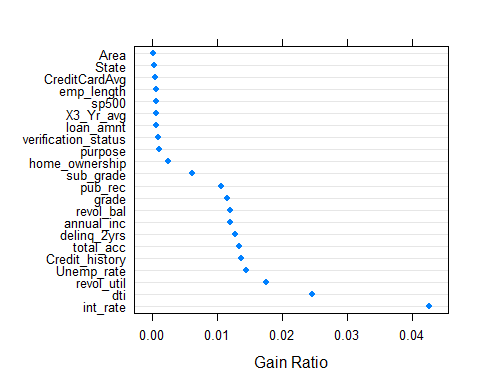
save(boruta.train, file = 'C:\\Ryerson Capstone\\data\\LC\\boruta.RData')

Feature evaluation using the CORElear package.

library(CORElearn)  
  
Gini <- attrEval(factor(loan\_status) ~ ., select(train, -status\_3way), estimator="Gini")  
GainRatio <- attrEval(factor(loan\_status) ~ ., select(train, -status\_3way), estimator="GainRatio")  
dotplot(sort(Gini, decreasing = TRUE), xlab = 'Gini Index')



dotplot(sort(GainRatio, decreasing = TRUE), xlab = 'Gain Ratio')

 Unfortunately, we don’t have strong predictors for the target variables among our independent features. Moreover, the best predictors come from the parameters set by Lending Club: grade, subgrade, and interest rate. Since state is a week predictor it will be dropped and keep only the geographic are. Also, because the grade and subgrade are correlated will focus on subgrade.

# References

Cohen, Maxime C., Guetta, C. Daniel, Jiao, Kevin, & Provost, Foster. (2018). Data-Driven Investment Strategies for Peer-to-Peer Lending: A Case Study for Teaching Data Science. Big Data, 6(3), 191-213. doi: 10.1089/big.2018.0092