Data Cleaning LCdata

Group1

2022-12-29

Contents

Abstract

Data cleaning is an essential preprocessing step in the data analysis process. The LCData dataset, from the US-based crowdlender LoanClear, is a large dataset that requires extensive cleaning due to the presence of many missing values (NA's) and characters. The ultimate goal of this task is to create a model that can accurately predict interest rates. To achieve this, a thorough data cleaning process will be necessary to ensure that the data is accurate and ready for analysis. This may involve identifying and correcting errors, filling in missing values, and removing any unnecessary or irrelevant data. By completing this data cleaning task, we can better understand the underlying trends and patterns in the data, and use these insights to develop a more effective model for predicting interest rates.

```
getwd()
```

```
## [1] "C:/Users/yanni/OneDrive/Dokumente/FHNW_Data_Science/Scripts"
```

```
cleaning <- read.csv("../Data/In/Project/LCdata.csv", row.names=NULL,sep = ";" )
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

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

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

```
library(ggplot2)
summary(cleaning)
```

```
##        id              member_id          loan_amnt      funded_amnt
##  Min.   :   54734   Min.   :   70473   Min.   :  500   Min.   :  500
##  1st Qu.: 9207230   1st Qu.:10877939   1st Qu.: 8000   1st Qu.: 8000
##  Median :34433372   Median :37095300   Median :13000   Median :13000
##  Mean   :32463636   Mean   :35000265   Mean   :14754   Mean   :14741
##  3rd Qu.:54900100   3rd Qu.:58470266   3rd Qu.:20000   3rd Qu.:20000
##  Max.   :68617057   Max.   :73544841   Max.   :35000   Max.   :35000
##
##  funded_amnt_inv     term              int_rate      installment
##  Min.   :    0   Length:798641      Min.   : 5.32   Min.   :  15.67
##  1st Qu.: 8000   Class :character   1st Qu.: 9.99   1st Qu.: 260.55
##  Median :13000   Mode  :character   Median :12.99   Median : 382.55
##  Mean   :14702                      Mean   :13.24   Mean   : 436.66
##  3rd Qu.:20000                      3rd Qu.:16.20   3rd Qu.: 572.60
##  Max.   :35000                      Max.   :28.99   Max.   :1445.46
##
##   emp_title          emp_length         home_ownership       annual_inc
##  Length:798641      Length:798641      Length:798641      Min.   :      0
##  Class :character   Class :character   Class :character   1st Qu.:  45000
```

```
##  Mode  :character   Mode  :character   Mode  :character   Median :  65000
##                                                           Mean   :  75014
##                                                           3rd Qu.:  90000
##                                                           Max.   :9500000
##                                                           NA's   :4
##  verification_status   issue_d          loan_status         pymnt_plan
##  Length:798641       Length:798641      Length:798641      Length:798641
##  Class :character    Class :character   Class :character   Class :character
##  Mode  :character    Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##      url                desc              purpose             title
##  Length:798641      Length:798641      Length:798641      Length:798641
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    zip_code          addr_state             dti           delinq_2yrs
##  Length:798641      Length:798641      Min.   :   0.00   Min.   : 0.0000
##  Class :character   Class :character   1st Qu.:  11.91   1st Qu.: 0.0000
##  Mode  :character   Mode  :character   Median :  17.66   Median : 0.0000
##                                        Mean   :  18.16   Mean   : 0.3145
##                                        3rd Qu.:  23.95   3rd Qu.: 0.0000
##                                        Max.   :9999.00   Max.   :39.0000
##                                                          NA's   :25
##  earliest_cr_line   inq_last_6mths    mths_since_last_delinq
##  Length:798641      Min.   : 0.0000   Min.   :  0.0
##  Class :character   1st Qu.: 0.0000   1st Qu.: 15.0
##  Mode  :character   Median : 0.0000   Median : 31.0
##                     Mean   : 0.6947   Mean   : 34.1
##                     3rd Qu.: 1.0000   3rd Qu.: 50.0
##                     Max.   :33.0000   Max.   :188.0
##                     NA's   :25        NA's   :408818
##  mths_since_last_record    open_acc        pub_rec          revol_bal
##  Min.   :  0.0          Min.   : 0.00   Min.   : 0.0000   Min.   :      0
##  1st Qu.: 51.0          1st Qu.: 8.00   1st Qu.: 0.0000   1st Qu.:   6443
##  Median : 70.0          Median :11.00   Median : 0.0000   Median :  11876
##  Mean   : 70.1          Mean   :11.55   Mean   : 0.1953   Mean   :  16930
##  3rd Qu.: 92.0          3rd Qu.:14.00   3rd Qu.: 0.0000   3rd Qu.:  20839
##  Max.   :129.0          Max.   :90.00   Max.   :63.0000   Max.   :2904836
##  NA's   :675190         NA's   :25      NA's   :25        NA's   :2
##    revol_util       total_acc      initial_list_status   out_prncp
##  Min.   :  0.00   Min.   :  1.00   Length:798641       Min.   :    0
##  1st Qu.: 37.70   1st Qu.: 17.00   Class :character    1st Qu.:    0
##  Median : 56.00   Median : 24.00   Mode  :character    Median : 6454
##  Mean   : 55.05   Mean   : 25.27                       Mean   : 8402
##  3rd Qu.: 73.50   3rd Qu.: 32.00                       3rd Qu.:13661
##  Max.   :892.30   Max.   :169.00                       Max.   :49373
##  NA's   :454      NA's   :25
##  out_prncp_inv    total_pymnt    total_pymnt_inv total_rec_prncp
```

```
##  Min.   :    0   Min.   :    0   Min.   :    0   Min.   :    0
##  1st Qu.:    0   1st Qu.: 1913   1st Qu.: 1898   1st Qu.: 1200
##  Median : 6452   Median : 4895   Median : 4862   Median : 3216
##  Mean   : 8399   Mean   : 7557   Mean   : 7520   Mean   : 5757
##  3rd Qu.:13656   3rd Qu.:10612   3rd Qu.:10561   3rd Qu.: 8000
##  Max.   :49373   Max.   :56809   Max.   :56475   Max.   :35000
##
##  total_rec_int     total_rec_late_fee   recoveries
##  Min.   :    0.0   Min.   :  0.0000   Min.   :    0.00
##  1st Qu.:  441.5   1st Qu.:  0.0000   1st Qu.:    0.00
##  Median : 1072.7   Median :  0.0000   Median :    0.00
##  Mean   : 1753.8   Mean   :  0.3962   Mean   :   45.88
##  3rd Qu.: 2236.9   3rd Qu.:  0.0000   3rd Qu.:    0.00
##  Max.   :24205.6   Max.   :358.6800   Max.   :33520.27
##
##  collection_recovery_fee last_pymnt_d       last_pymnt_amnt
##  Min.   :   0.000        Length:798641      Min.   :    0.0
##  1st Qu.:   0.000        Class :character   1st Qu.:  279.9
##  Median :   0.000        Mode  :character   Median :  462.6
##  Mean   :   4.874                           Mean   : 2162.3
##  3rd Qu.:   0.000                           3rd Qu.:  830.3
##  Max.   :7002.190                           Max.   :36475.6
##
##  next_pymnt_d       last_credit_pull_d collections_12_mths_ex_med
##  Length:798641      Length:798641      Min.   : 0.00000
##  Class :character   Class :character   1st Qu.: 0.00000
##  Mode  :character   Mode  :character   Median : 0.00000
##                                        Mean   : 0.01447
##                                        3rd Qu.: 0.00000
##                                        Max.   :20.00000
##                                        NA's   :126
##  mths_since_last_major_derog  policy_code application_type   annual_inc_joint
##  Min.   :  0.0             Min.   :1    Length:798641      Min.   : 17950
##  1st Qu.: 27.0             1st Qu.:1    Class :character   1st Qu.: 76167
##  Median : 44.0             Median :1    Mode  :character   Median :101886
##  Mean   : 44.1             Mean   :1                       Mean   :110745
##  3rd Qu.: 61.0             3rd Qu.:1                       3rd Qu.:133000
##  Max.   :188.0             Max.   :1                       Max.   :500000
##  NA's   :599107                                            NA's   :798181
##    dti_joint      verification_status_joint acc_now_delinq
##  Min.   : 3.0     Length:798641             Min.   : 0.000000
##  1st Qu.:13.3     Class :character          1st Qu.: 0.000000
##  Median :17.7     Mode  :character          Median : 0.000000
##  Mean   :18.4                               Mean   : 0.005026
##  3rd Qu.:22.6                               3rd Qu.: 0.000000
##  Max.   :43.9                               Max.   :14.000000
##  NA's   :798183                             NA's   :25
##   tot_coll_amt      tot_cur_bal       open_acc_6m       open_il_6m
##  Min.   :      0   Min.   :      0   Min.   : 0.0     Min.   : 0.0
##  1st Qu.:      0   1st Qu.:  29861   1st Qu.: 0.0     1st Qu.: 1.0
##  Median :      0   Median :  80647   Median : 1.0     Median : 2.0
##  Mean   :    228   Mean   : 139508   Mean   : 1.1     Mean   : 2.9
##  3rd Qu.:      0   3rd Qu.: 208229   3rd Qu.: 2.0     3rd Qu.: 4.0
##  Max.   :9152545   Max.   :8000078   Max.   :14.0     Max.   :33.0
```

```
##  NA's   :63276     NA's   :63276     NA's   :779525   NA's   :779525
##   open_il_12m      open_il_24m     mths_since_rcnt_il  total_bal_il
##  Min.   : 0.0     Min.   : 0.0     Min.   :  0.0      Min.   :     0
##  1st Qu.: 0.0     1st Qu.: 0.0     1st Qu.:  6.0      1st Qu.: 10164
##  Median : 0.0     Median : 1.0     Median : 12.0      Median : 24545
##  Mean   : 0.8     Mean   : 1.7     Mean   : 21.1      Mean   : 36429
##  3rd Qu.: 1.0     3rd Qu.: 2.0     3rd Qu.: 23.0      3rd Qu.: 47640
##  Max.   :12.0     Max.   :19.0     Max.   :363.0      Max.   :878459
##  NA's   :779525   NA's   :779525   NA's   :780030     NA's   :779525
##     il_util        open_rv_12m      open_rv_24m       max_bal_bc
##  Min.   :  0.0    Min.   : 0.0     Min.   : 0       Min.   :    0
##  1st Qu.: 58.4    1st Qu.: 0.0     1st Qu.: 1       1st Qu.: 2406
##  Median : 74.8    Median : 1.0     Median : 2       Median : 4502
##  Mean   : 71.5    Mean   : 1.4     Mean   : 3       Mean   : 5878
##  3rd Qu.: 87.7    3rd Qu.: 2.0     3rd Qu.: 4       3rd Qu.: 7774
##  Max.   :223.3    Max.   :22.0     Max.   :43       Max.   :83047
##  NA's   :782007   NA's   :779525   NA's   :779525   NA's   :779525
##     all_util     total_rev_hi_lim      inq_fi        total_cu_tl
##  Min.   :  0.0    Min.   :      0   Min.   : 0.0     Min.   : 0.0
##  1st Qu.: 47.6    1st Qu.:  13900   1st Qu.: 0.0     1st Qu.: 0.0
##  Median : 61.9    Median :  23700   Median : 0.0     Median : 0.0
##  Mean   : 60.8    Mean   :  32093   Mean   : 0.9     Mean   : 1.5
##  3rd Qu.: 75.2    3rd Qu.:  39800   3rd Qu.: 1.0     3rd Qu.: 2.0
##  Max.   :151.4    Max.   :9999999   Max.   :16.0     Max.   :35.0
##  NA's   :779525   NA's   :63276     NA's   :779525   NA's   :779525
##   inq_last_12m
##  Min.   :-4
##  1st Qu.: 0
##  Median : 2
##  Mean   : 2
##  3rd Qu.: 3
##  Max.   :32
##  NA's   :779525
```

NA - Cleaning

To locate rows in a specific column containing NA values, you can use the which() function in conjunction with the dplyr library. To use this library, you can press Alt+Shift+M to call it. This library is widely used and can be easily found by searching for it. By selecting the appropriate column and adding the argument TRUE to the which() function, you can identify the rows containing NA values.

During the data cleaning process, I chose to use the dplyr library to select the rows containing annual income data. I utilized the filter() function to remove all rows containing NA values in the annual income column, as there were only a small number of such rows. I then used the select() function to delete entire columns. By preceding the column names with a minus sign, I specified which columns to delete.

The mutate() function allows for the creation of new variables while preserving existing ones. In this case, I created a new column called _cat. The ifelse() function was then used to transform the months since delinq data into the _cat column. By inspecting the months since delinq data in a histogram, I observed that it ranged up to 500 months. Grouping this data was a subjective process that required business knowledge.

After grouping the data with the ifelse() function, it was necessary to convert the resulting categories into numeric values using the mutate() function. This allowed for further analysis and manipulation of the data.

Delete columns

```
which(is.na(cleaning$annual_inc)== TRUE)
```

```
## [1]     2     3 44689 73832
```

```
library(dplyr)

cleaning <- cleaning %>%
  filter(!(is.na(annual_inc))) %>%
    filter(!(is.na(delinq_2yrs)))%>%
      filter(!(is.na(revol_bal))) %>%
        filter(!(is.na(revol_util))) %>%
          filter(!(is.na(collections_12_mths_ex_med))) %>%

select( -id, -member_id, -title, -emp_title, -loan_status, -funded_amnt, -funded_amnt_inv, -loan_status

  mutate(
    mths_since_delinq_cat = ifelse(is.na(mths_since_last_delinq)== TRUE,"No_delinq",
                               ifelse(mths_since_last_delinq <= 12, "recent",
                                  ifelse(mths_since_last_delinq <= 36, "1_to_3_years",
                                         ifelse(mths_since_last_delinq <= 60,     "3_to_5_years","more_

  ) %>% select(-mths_since_last_delinq)

cleaning$mths_since_delinq_cat <- as.factor(cleaning$mths_since_delinq_cat)
```

The initial step in the data cleaning process involved the removal of NA values and the transformation of the data into a more manageable format. This provided a solid foundation for subsequent steps in the cleaning process.

One column, delinq_2_years, contained 21 NA values. The values in this column ranged from 0 to 39, indicating the number of "bad entries" in a particular register. The question then arose as to how to handle the NA values in this column: should the entire row be deleted, or the entire column? Most of the cases in the dataset had no delinquency within the last two years, so the impact on the overall analysis of the few cases with delinquency needed to be considered.

The revol_bal column contained only 2 NA values, which could be easily removed by deleting the corresponding rows. The revol_util column, on the other hand, contained 429 NA values, which represented a relatively small proportion of the overall dataset of 800,000 entries. Similarly, the collections_12_mths_ex_med column contained 101 NA values, which could also be considered negligible in relation to the size of the dataset.

Summary of NA's

Now to the cases that have more than 1k NA's which should not be deleted, are the following:

mths_since_last_record 675165 The number of months since the last public record. mths_since_last_major_derog 599082 Months since most recent 90-day or worse rating

annual_inc_joint 798156 The combined self-reported annual income provided by the co-borrowers during registration
dti_joint 798158 A ratio calculated using the co-borrowers' total monthly payments on the total debt obligations, excluding mortgages and the requested LC loan, divided by the co-borrowers' combined self-reported monthly income

tot_coll_amt 63251 Total collection amounts ever owed

tot_cur_bal 63251 Total current balance of all accounts

open_acc_6m 779500 Number of open trades in last 6 months

open_il_6m 779500 Number of currently active installment trades

open_il_12m 779500 Number of installment accounts opened in past 12 months open_il_24m 779500 Number of installment accounts opened in past 24 months mths_since_rcnt_il 780005 Months since most recent installment accounts opened

total_bal_il 779500 Total current balance of all installment accounts

il_util 781982 Ratio of total current balance to high credit/credit limit on all install acct

open_rv_12m 779500 Number of revolving trades opened in past 12 months

open_rv_24m 779500
total_rev_hi_lim 63251 Total revolving high credit/credit limit

max_bal_bc 779500 Maximum current balance owed on all revolving

accounts all_util 779500 Balance to credit limit on all trades

inq_fi 779500 Number of personal finance inquiries

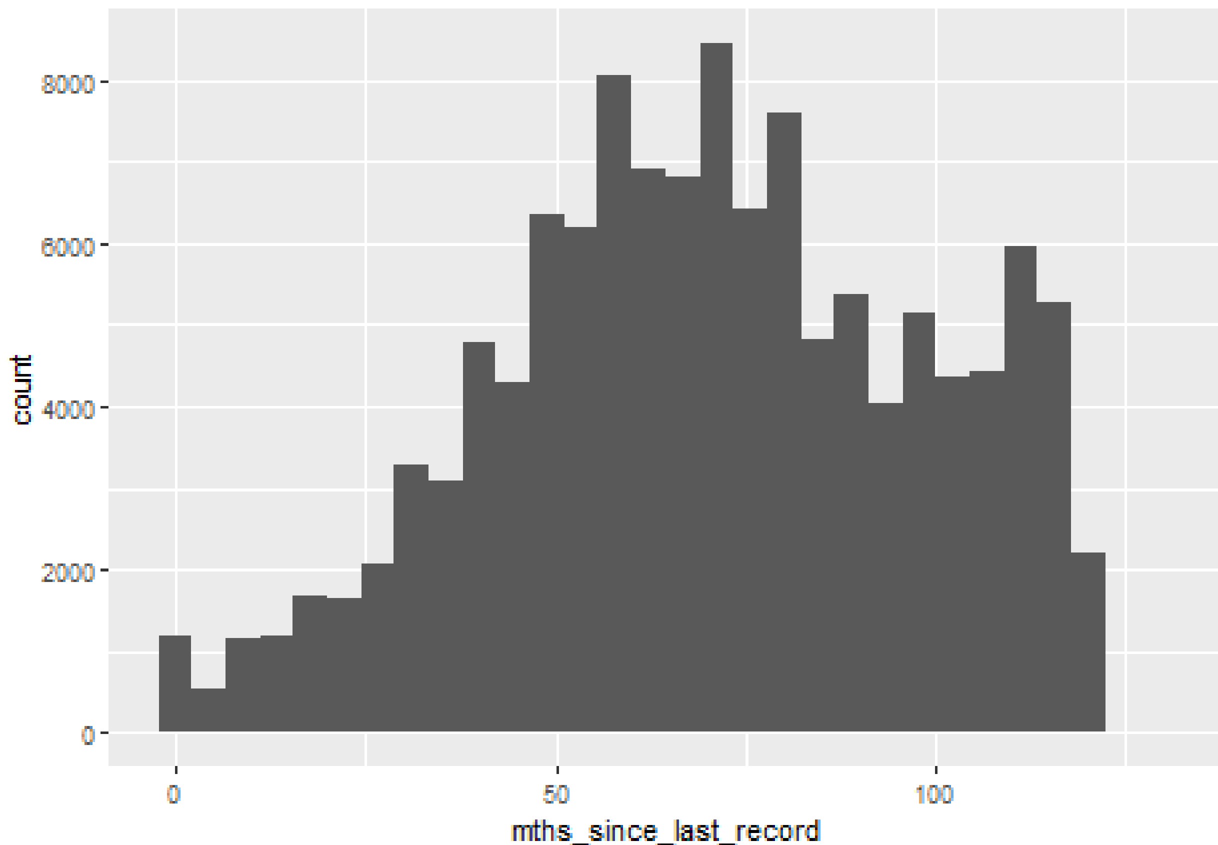
total_cu_tl 779500 Number of finance trades

inq_last_12m 779500 Number of credit inquiries in past 12 months

```
ggplot(data = cleaning, mapping = aes(x=mths_since_last_record))+geom_histogram()
```

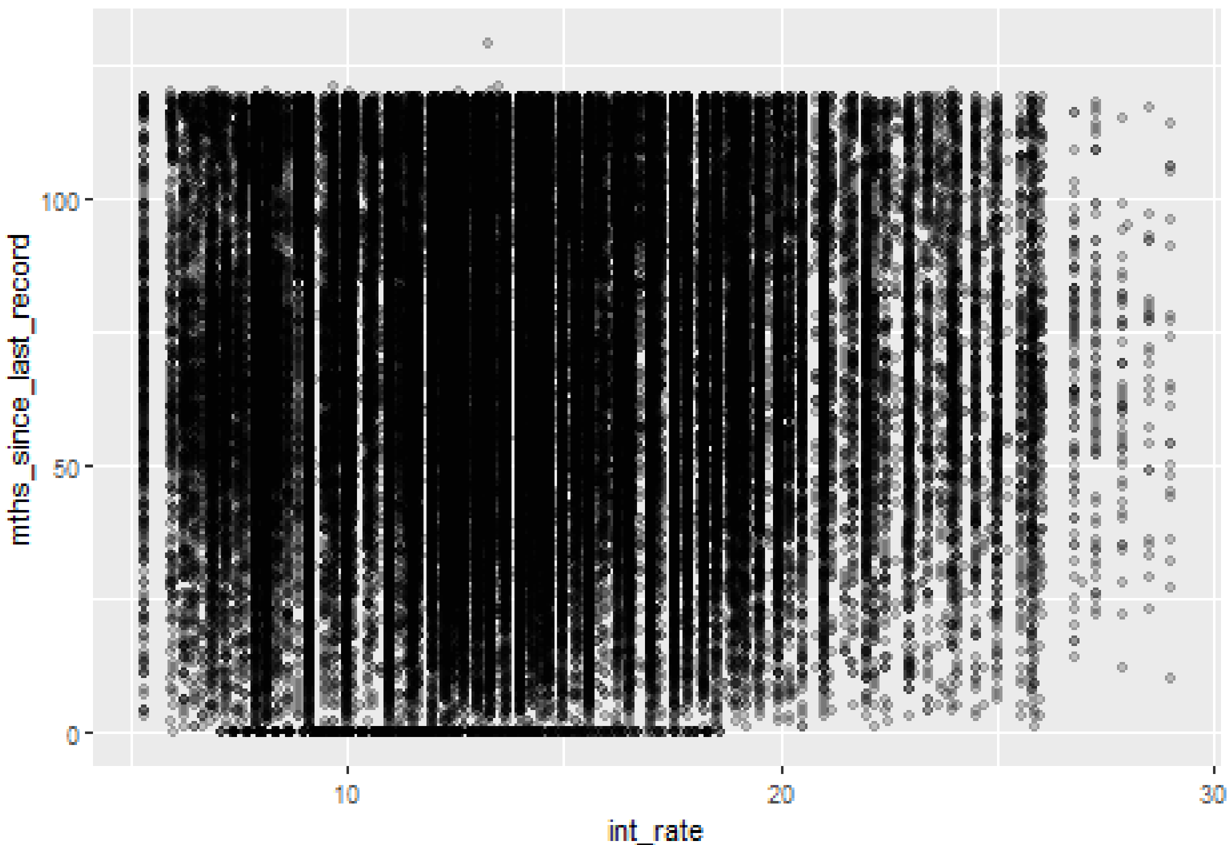
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 674745 rows containing non-finite values (`stat_bin()`).
```




```
ggplot(data = cleaning, mapping = aes(x=int_rate,y=mths_since_last_record))+geom_point(alpha=0.2)
```

```
## Warning: Removed 674745 rows containing missing values (`geom_point()`).
```

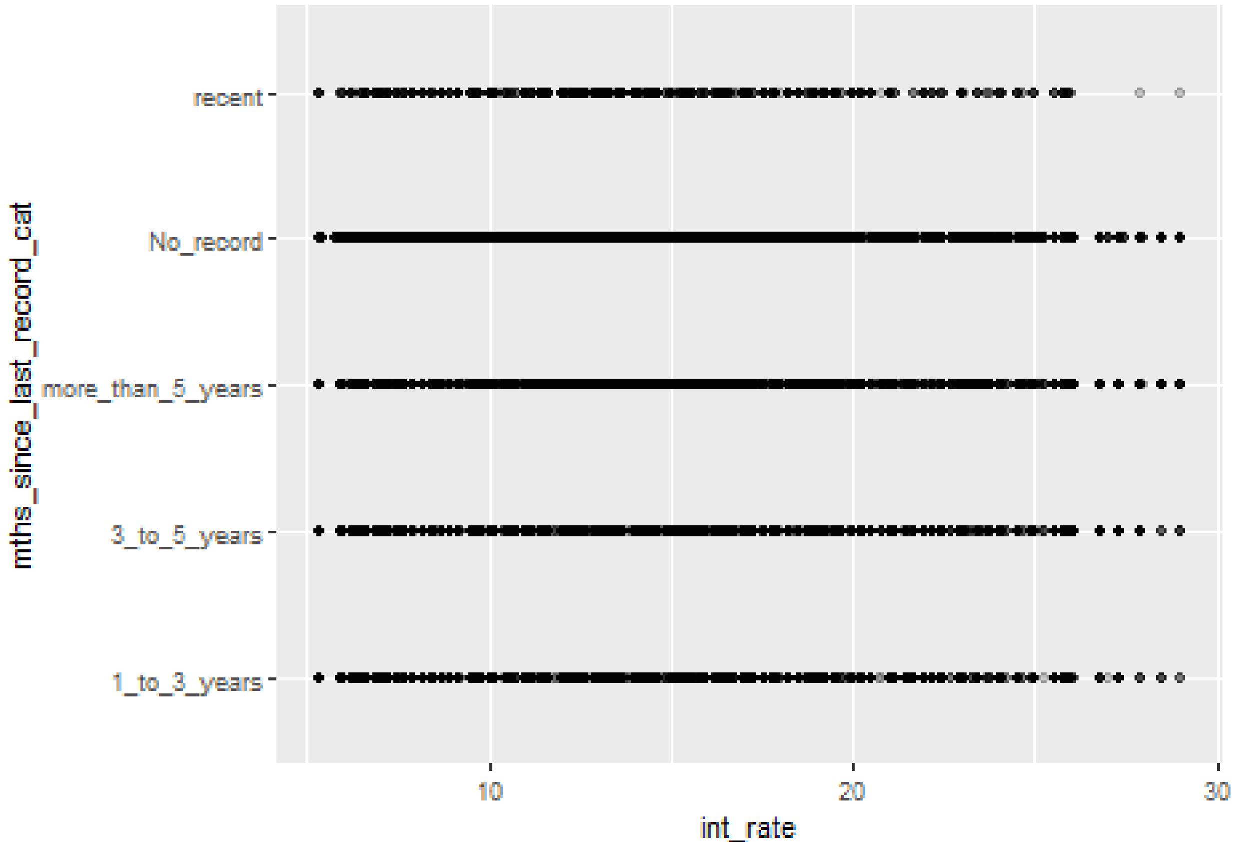


After visualizing the data through plotting, no significant correlation was observed. As a result, an alternative approach to analyzing the data may be to try categorizing the variables in order to identify any potential patterns or trends. This method of analysis involves dividing the data into discrete groups or categories based on certain characteristics or attributes, and can be useful for identifying relationships between variables that may not be immediately apparent through other means. It is important to carefully consider the chosen criteria for categorization and to ensure that the resulting categories are meaningful and relevant to the research question at hand.

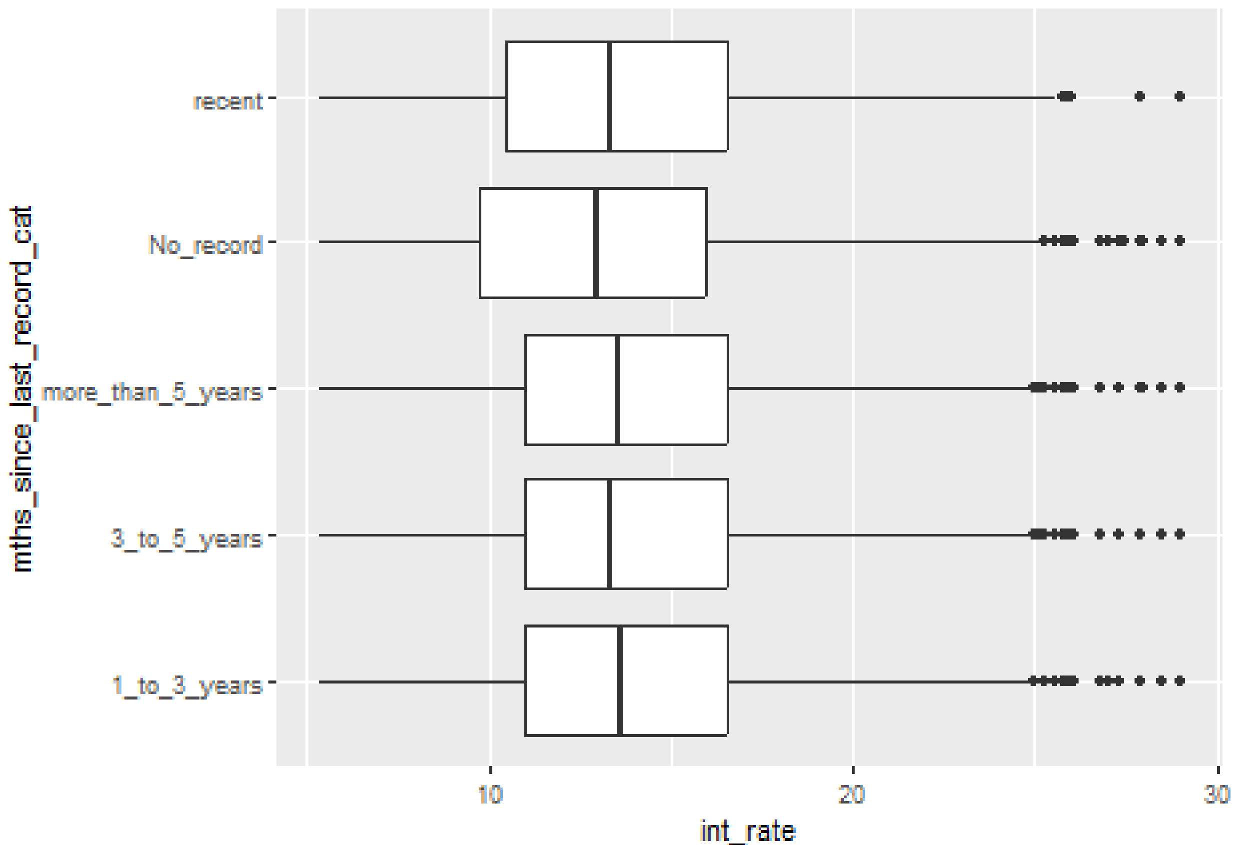
Cleaning of mths_since_last_record

```
#cleaning aproach for mths_since_last_record: These NA's seem to never have had a record in a debt enfo

cleaning <- cleaning %>% mutate(mths_since_last_record = ifelse(is.na(mths_since_last_record), 0, mths_s

cleaning <- cleaning %>%
  mutate(mths_since_last_record_cat = ifelse(mths_since_last_record== 0,"No_record",
                                ifelse(mths_since_last_record <= 12, "recent",
                                   ifelse(mths_since_last_record <= 36, "1_to_3_years",
                                          ifelse(mths_since_last_record <= 60,     "3_to_5_years","more_

cleaning$mths_since_last_record_cat <- as.factor(cleaning$mths_since_last_record_cat)

#Plotting again to see results
```

```
ggplot(data = cleaning, mapping = aes(x=int_rate,y=mths_since_last_record_cat))+geom_point(alpha=0.2)
```




```
ggplot(data = cleaning, mapping = aes(x=int_rate,y=mths_since_last_record_cat))+geom_boxplot()
```

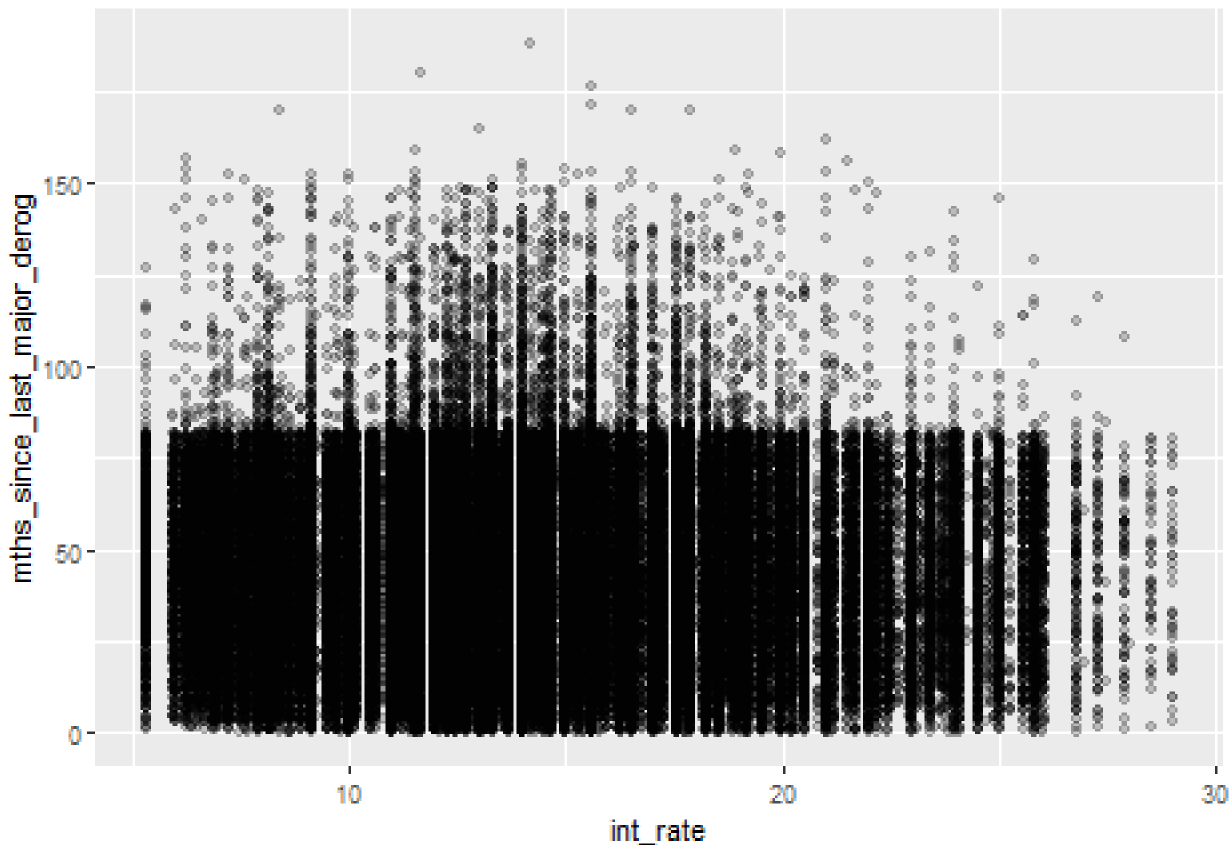


The results of the analysis are surprising, as there is only a minor and insignificant difference in the interest rate between individuals with no entries in a public register and those with a record of negative entries. This suggests that the presence or absence of such entries has little impact on the interest rate, and raises questions about the underwriting practices of Lending Club. It is possible that these factors are not being properly considered in the underwriting process, which may contribute to the company's current status. Further investigation may be necessary to understand the underlying causes of these unexpected results.

Cleaning of mths__since__last__major__derog

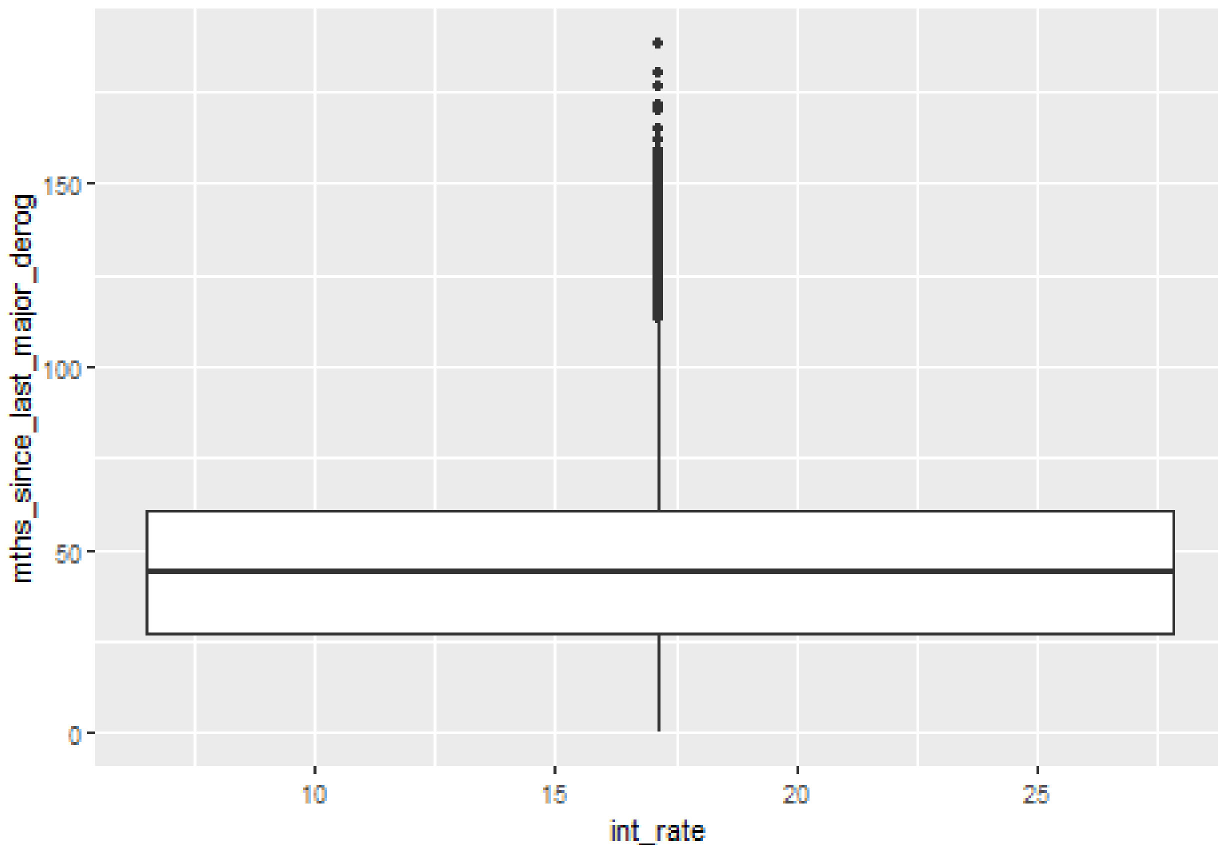
```
#Plotting uncleaned mths_since_last_major_derog
ggplot(data = cleaning, mapping = aes(x=int_rate,y=mths_since_last_major_derog))+geom_point(alpha=0.2)
```

```
## Warning: Removed 598693 rows containing missing values (‘geom_point()‘).
```

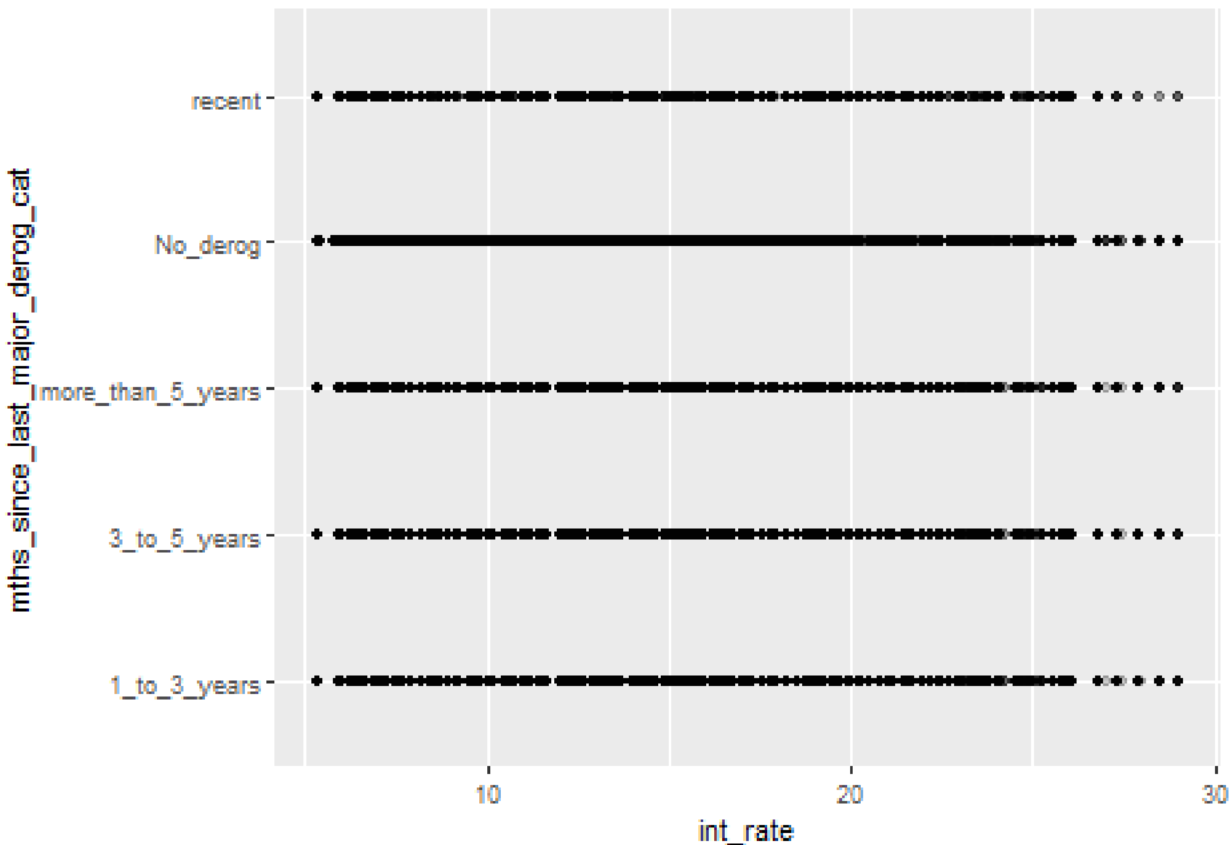



```
ggplot(data = cleaning, mapping = aes(x=int_rate,y=mths_since_last_major_derog))+geom_boxplot()
```

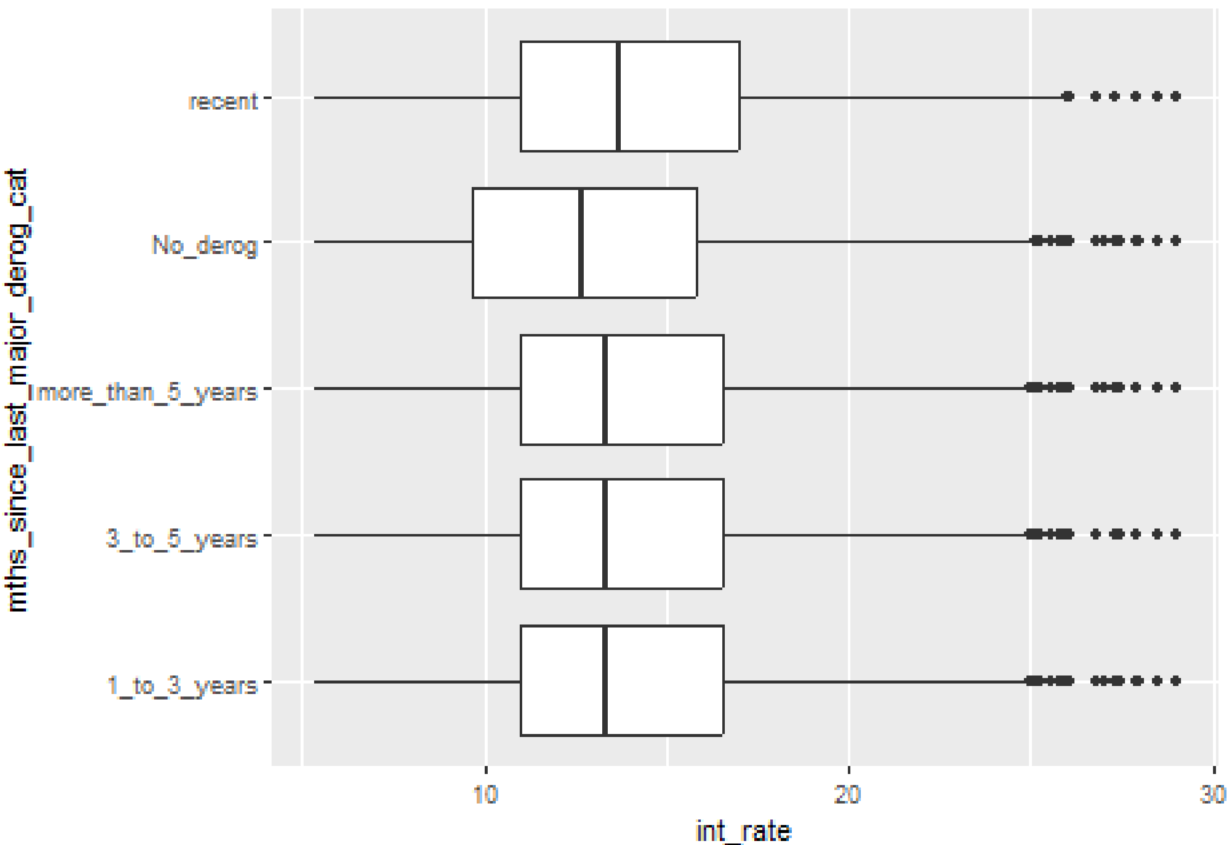
```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?

## Warning: Removed 598693 rows containing non-finite values (`stat_boxplot()`).
```




```
#Cleaning mths_since_last_major_derog
cleaning <- cleaning %>% mutate(
    mths_since_last_major_derog_cat = ifelse(is.na(mths_since_last_major_derog)== TRUE,"No_derog",
                                ifelse(mths_since_last_major_derog <= 12, "recent",
                                   ifelse(mths_since_last_major_derog <= 36, "1_to_3_years",
                                          ifelse(mths_since_last_major_derog <= 60,     "3_to_5_years",

  ) %>% select(-mths_since_last_major_derog)

cleaning$mths_since_last_major_derog_cat <- as.factor(cleaning$mths_since_last_major_derog_cat)

#Plotting cleaned mths_since_last_major_derog

ggplot(data = cleaning, mapping = aes(x=int_rate,y=mths_since_last_major_derog_cat))+geom_point(alpha=0
```




```
ggplot(data = cleaning, mapping = aes(x=int_rate,y=mths_since_last_major_derog_cat))+geom_boxplot()
```



The analysis of the "derog" variable yielded similar results to those obtained for the "last record" variable, in that there is little difference in the interest rate between individuals with different levels of derogatory entries. This suggests that the presence or absence of such entries has little impact on the interest rate, which raises concerns about the underwriting practices of Lending Club. The data appears to be unusual and may indicate poor underwriting processes, which could potentially contribute to the company's current status. Further investigation may be necessary to understand the reasons for these unexpected results.

Cleaning of annual_inc_joint and dti_joint

Before cleaning, the data only indicates whether there is a joint income present. This also applies to the "dti" and "dti_joint" variables. In order to accurately represent the income and debt-to-income ratio for each individual, it is necessary to merge the "dti" and "annual_inc" variables with their respective "joint" counterparts, depending on whether the application is individual or joint. This can be accomplished using the ifelse function in the mutate function, which allows us to specify conditions for replacing certain values with others. For example, we can use the ifelse function to replace any empty values in the "address" column with the corresponding value from the "work_address" column. It is important to carefully consider the chosen method for handling missing or incomplete data in order to ensure the accuracy and reliability of the cleaned dataset.

```
#merging annual income
cleaning <- cleaning %>% mutate(
    annual_inc_merged = ifelse(is.na(annual_inc_joint)== TRUE, annual_inc,annual_inc_joint))

cleaning <- cleaning %>% select(-annual_inc,-annual_inc_joint)
```

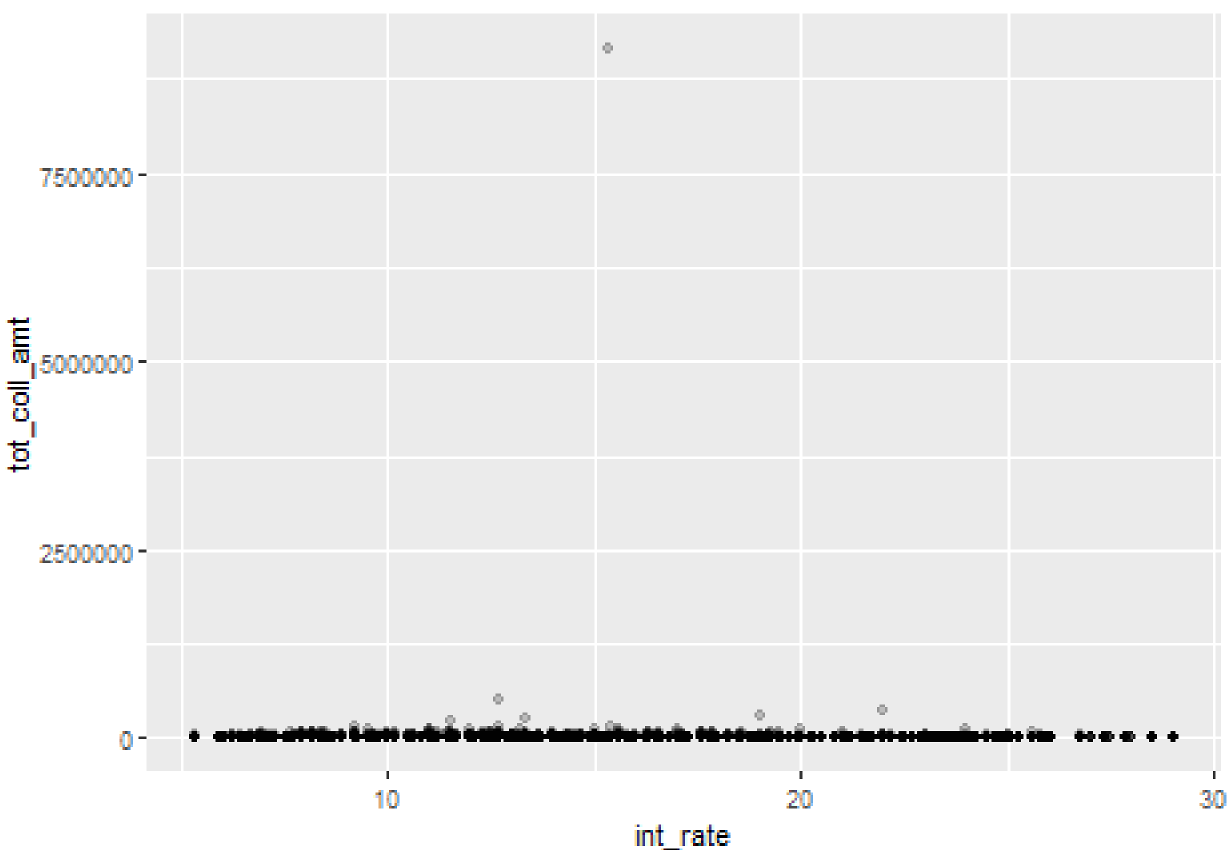
```r
#merging debt to income ratio
cleaning <- cleaning %>% mutate(
    dti_merged = ifelse(is.na(dti_joint)== TRUE, dti,dti_joint))

cleaning <- cleaning %>% select(-dti,-dti_joint)
```

Cleaning of tot_coll_amt There appears to be a correlation between

the interest rate and the total collateral amount, making it worthwhile to clean the relevant column in the dataset. There may be missing values, or "NA's," in this column, which may indicate that these customers either have no debt or do not have any debt when obtaining a loan from LoanClear. In order to accurately represent this information, it may be necessary to replace these missing values with the value "0" to indicate the absence of debt. This will allow for more accurate analysis of the relationship between the interest rate and the total collateral amount. It is important to carefully consider the chosen method for handling missing data in order to ensure the accuracy and reliability of the cleaned dataset.

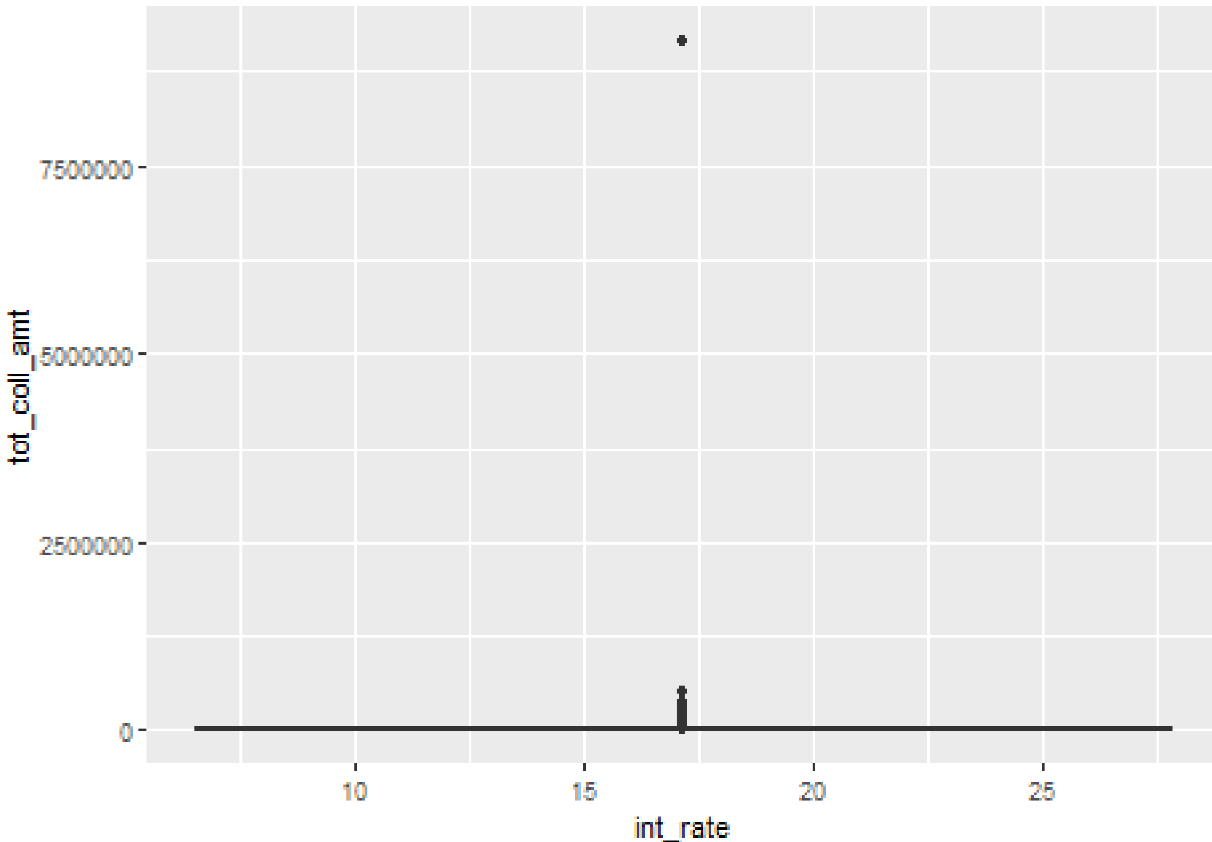
```r
#Plotting uncleaned tot_coll_amt

ggplot(data = cleaning, mapping = aes(x=int_rate,y=tot_coll_amt))+geom_point(alpha=0.2)
```

```
## Warning: Removed 63072 rows containing missing values (‘geom_point()‘).
```

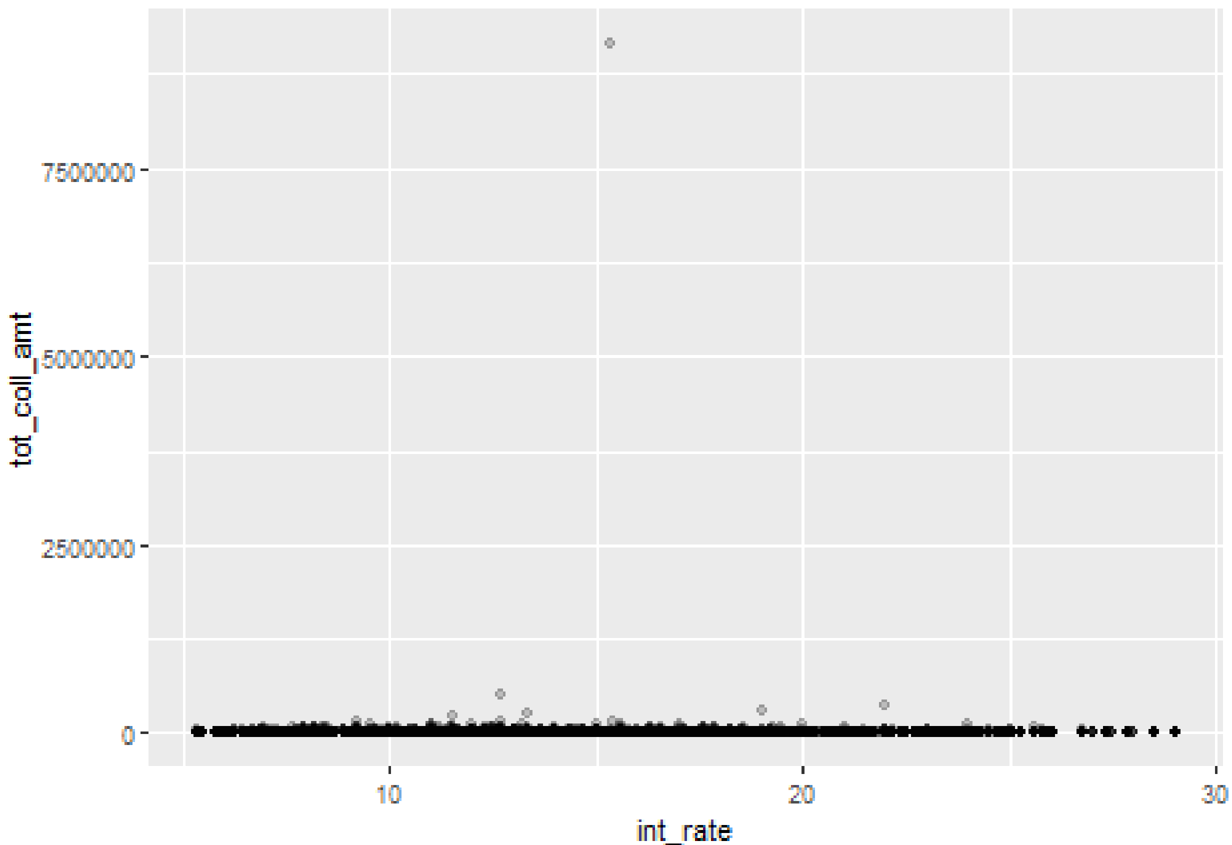


```r
ggplot(data = cleaning, mapping = aes(x=int_rate,y=tot_coll_amt))+geom_boxplot()
```

```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?

## Warning: Removed 63072 rows containing non-finite values (`stat_boxplot()`).
```

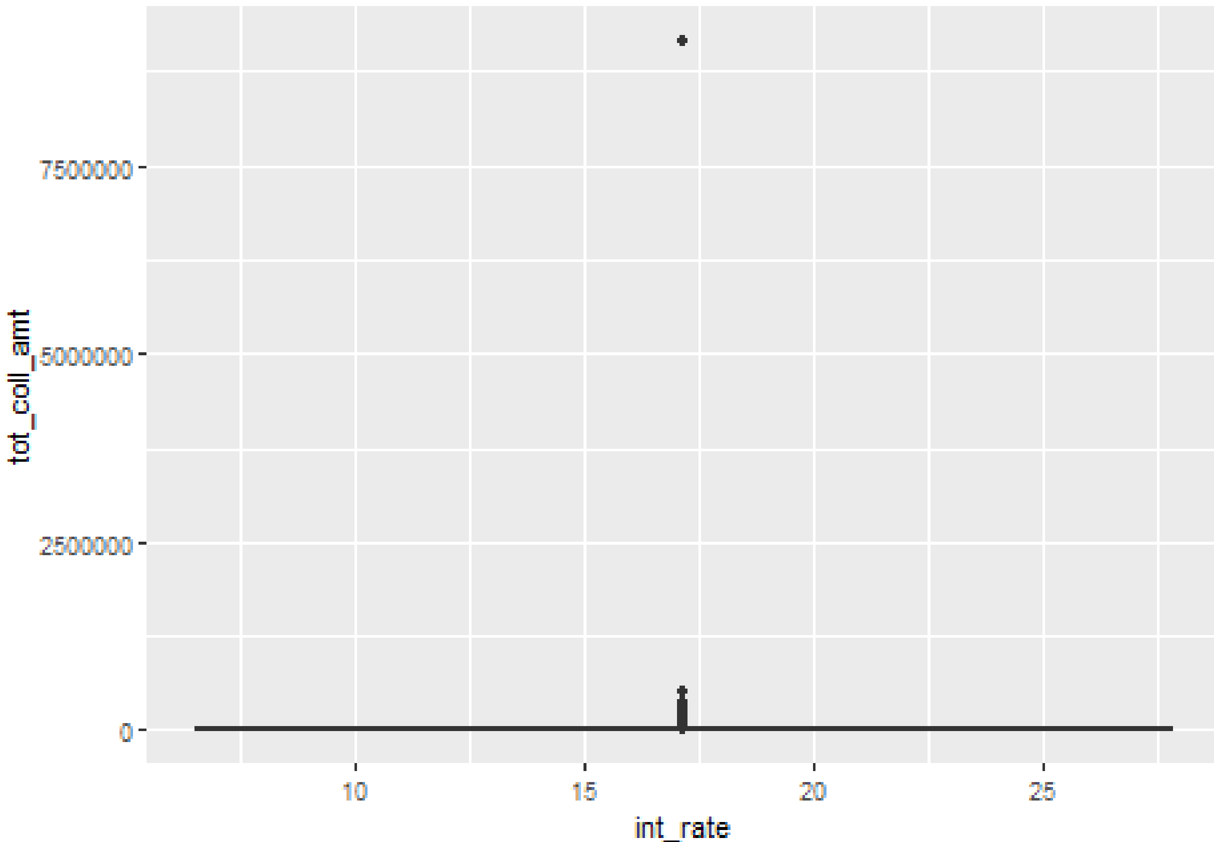



```r
#Cleaning tot_coll_amt
cleaning <- cleaning %>% mutate(
    tot_coll_amt = ifelse(is.na(tot_coll_amt)== TRUE,0, tot_coll_amt))


#Plotting cleaned tot_coll_amt

ggplot(data = cleaning, mapping = aes(x=int_rate,y=tot_coll_amt))+geom_point(alpha=0.2)
```




```
ggplot(data = cleaning, mapping = aes(x=int_rate,y=tot_coll_amt))+geom_boxplot()
```

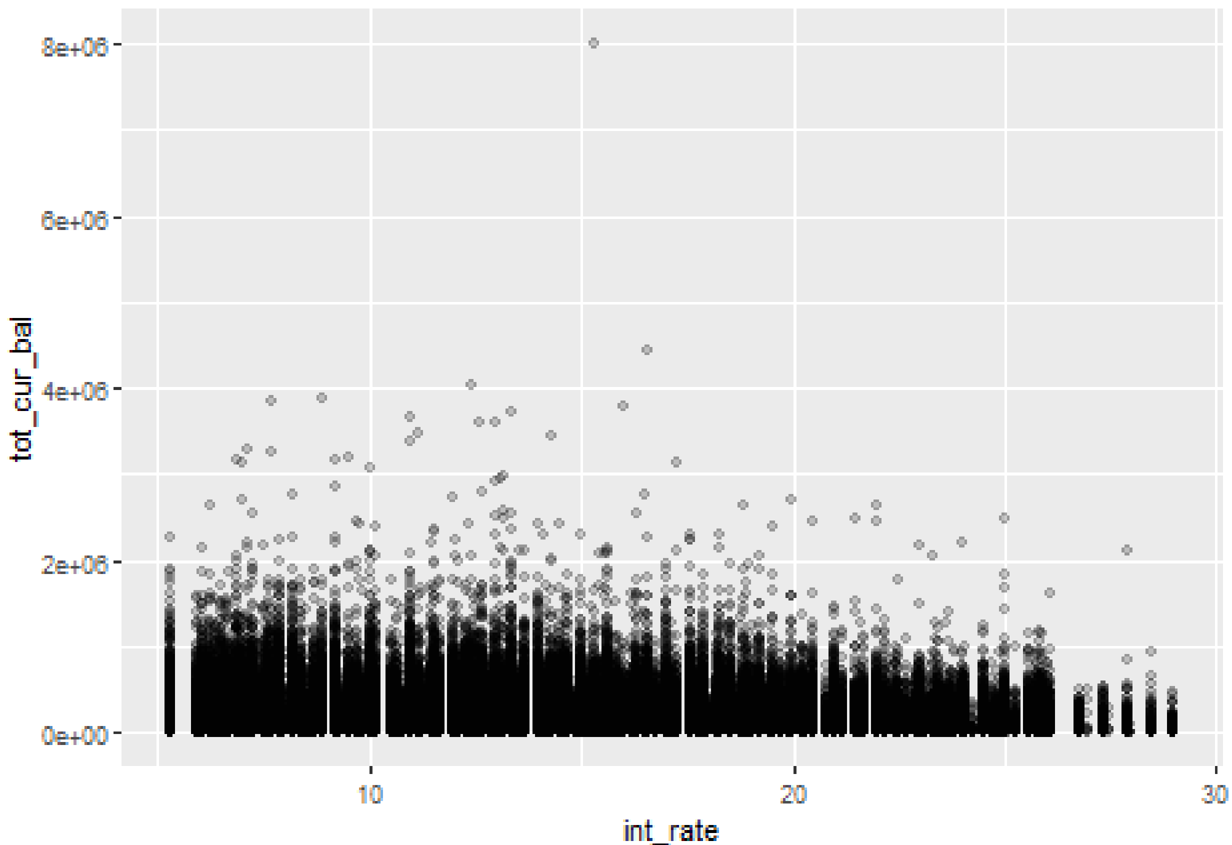
```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
```



Cleaning of tot__cur__bal Outliers here as well

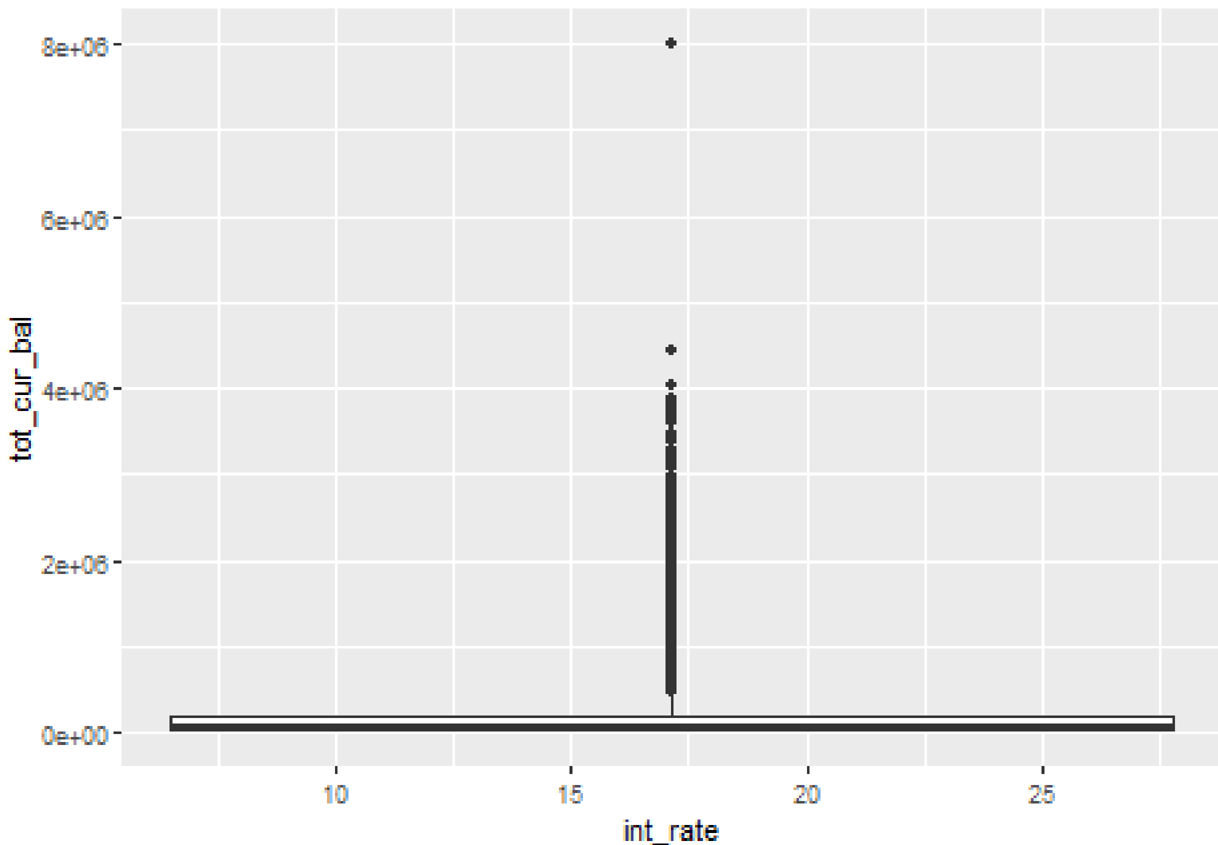
```
#Plotting uncleaned tot_cur_bal

ggplot(data = cleaning, mapping = aes(x=int_rate,y=tot_cur_bal))+geom_point(alpha=0.2)
```

```
## Warning: Removed 63072 rows containing missing values (`geom_point()`).
```

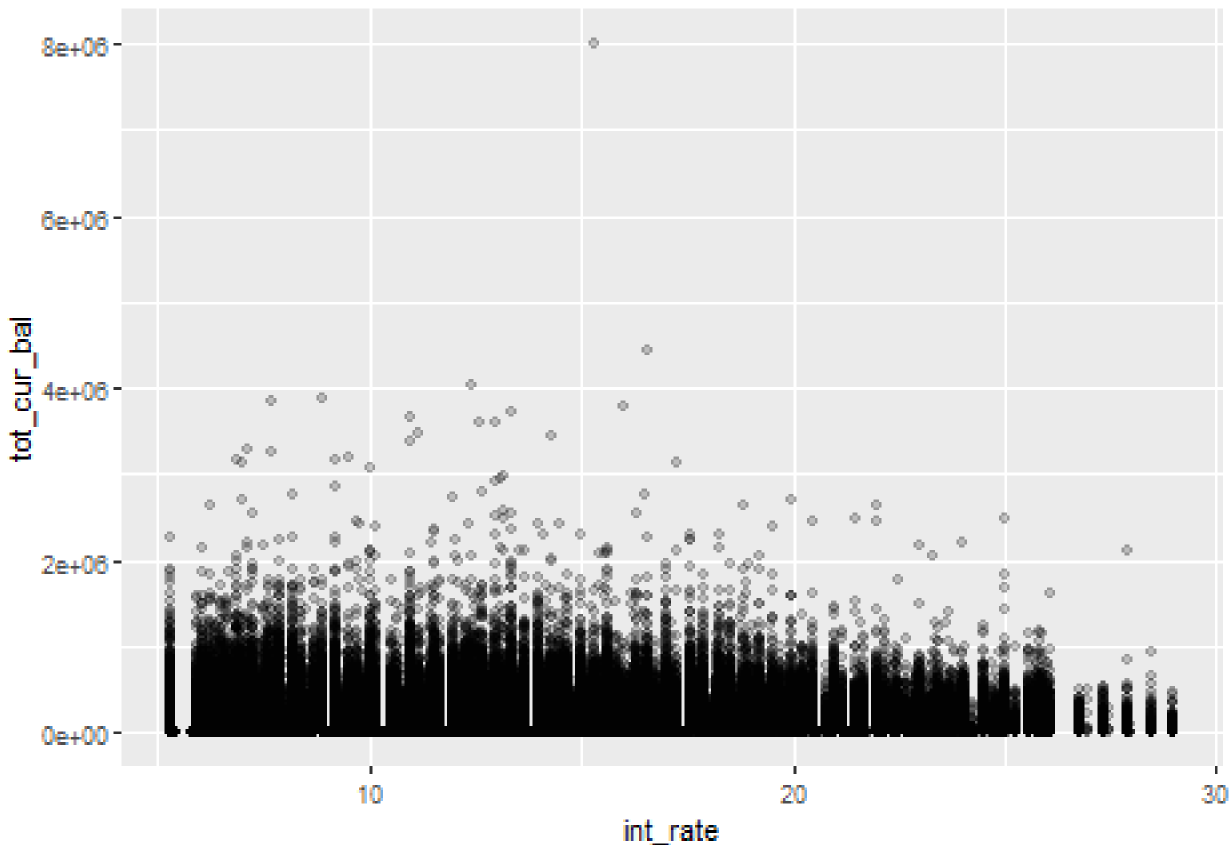



```r
ggplot(data = cleaning, mapping = aes(x=int_rate,y=tot_cur_bal))+geom_boxplot()
```

```
## Warning: Continuous x aesthetic
## i did you forget ‘aes(group = ...)‘?

## Warning: Removed 63072 rows containing non-finite values (‘stat_boxplot()‘).
```

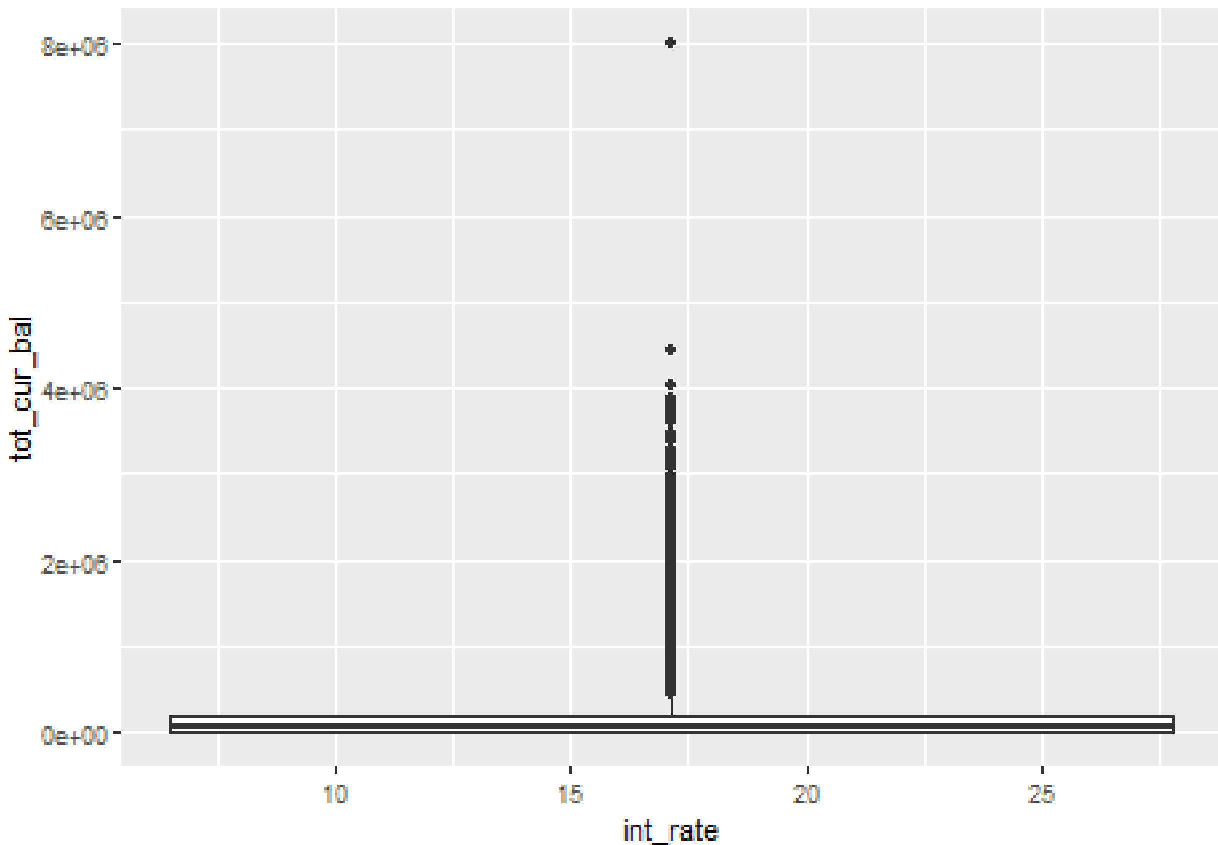



```
#Cleaning tot_cur_bal
cleaning <- cleaning %>% mutate(
    tot_cur_bal = ifelse(is.na(tot_cur_bal)== TRUE,0, tot_cur_bal))


#Plotting cleaned tot_cur_bal

ggplot(data = cleaning, mapping = aes(x=int_rate,y=tot_cur_bal))+geom_point(alpha=0.2)
```




```
ggplot(data = cleaning, mapping = aes(x=int_rate,y=tot_cur_bal))+geom_boxplot()
```

```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
```



Cleaning of open_acc_6m, open_il_6m, open_il_12m, open_il_24m,

mths_since_rcnt_il, total_bal_il, il_util, open_rv_12m, open_rv_24m, total_rev_hi_lim, max_bal_bc, all_util, inq_fi, total_cu_tl, inq_last_12m

```
cleaning <- cleaning %>%                                                mutate(
    open_acc_6m = ifelse(is.na(open_acc_6m)== TRUE,0, open_acc_6m)) %>% mutate(
    open_il_6m = ifelse(is.na(open_il_6m)== TRUE,0, open_il_6m))   %>% mutate(
    open_il_12m = ifelse(is.na(open_il_12m)== TRUE,0, open_il_12m)) %>% mutate(
    open_il_24m = ifelse(is.na(open_il_24m)== TRUE,0, open_il_24m)) %>% mutate(
    mths_since_rcnt_il = ifelse(is.na(mths_since_rcnt_il)== TRUE,0, mths_since_rcnt_il)) %>% mutate(
    total_bal_il = ifelse(is.na(total_bal_il)== TRUE,0, total_bal_il)) %>% mutate(
    il_util = ifelse(is.na(il_util)== TRUE,0, il_util)) %>% mutate(
    open_rv_12m = ifelse(is.na(open_rv_12m)== TRUE,0, open_rv_12m)) %>% mutate(
    total_rev_hi_lim = ifelse(is.na(total_rev_hi_lim)== TRUE,0, total_rev_hi_lim)) %>% mutate(
    max_bal_bc = ifelse(is.na(max_bal_bc)== TRUE,0, max_bal_bc)) %>% mutate(
    all_util = ifelse(is.na(all_util)== TRUE,0, all_util)) %>% mutate(
    inq_fi = ifelse(is.na(inq_fi)== TRUE,0, inq_fi)) %>% mutate(
    total_cu_tl = ifelse(is.na(total_cu_tl)== TRUE,0, total_cu_tl)) %>% mutate(
    inq_last_12m = ifelse(is.na(inq_last_12m)== TRUE,0, inq_last_12m)) %>% mutate(
    open_rv_24m = ifelse(is.na(open_rv_24m)== TRUE,0, open_rv_24m))
```

Changing characters to factors In data cleaning, it is often necessary

to convert variables that contain characters into a "factor" data type. Factors are a special data type in R that are used to represent categorical variables, which are variables that can take on a limited number of values. Factors are particularly useful when working with data that contains text values, such as "male" or "female," as they allow you to easily group and analyze the data based on these categories. When you "factorize" a column that contains characters, you are essentially creating a factor object from the character data, which allows you to more easily manipulate and analyze the data. Factors are typically created using the factor function in R, which allows you to specify the levels, or possible values, of the factor and assign a numerical value to each level. Factors are an important tool in data cleaning and analysis, as they allow you to more easily work with categorical data and draw meaningful conclusions from your data.

```
cleaning$verification_status <- as.factor(cleaning$verification_status)
cleaning$verification_status_joint <- as.factor(cleaning$verification_status_joint)
cleaning$application_type <- as.factor(cleaning$application_type)
cleaning$initial_list_status <- as.factor(cleaning$initial_list_status)
cleaning$term <- as.factor(cleaning$term)
cleaning$purpose <- as.factor(cleaning$purpose)
cleaning$emp_length <- as.factor(cleaning$emp_length)
```

"Upon reexamination of the summary, it is evident that there are only 460 joint applications, which represents a small subset of the total dataset containing approximately 800k rows. By merging the dti's, we are able to identify the data that is relevant to our research interests. Therefore, it is recommended to remove the columns verification_status_joint and application_type to avoid introducing unnecessary variability in our analysis."

```
cleaning <- cleaning %>% select(-verification_status_joint, -application_type)
```

Cleaning of emp_lenght and issue_d

In this code, we are examining a dataset called "cleaning" and performing some data cleaning and exploration. The first step is to identify the unique values in the "emp_length" column using the "unique" function. We then use the "filter" function to create a new dataset called "temp" that only includes rows where the "emp_length" value is "n/a."

Next, we use the "hist" function to create histograms of the "annual_inc_merged" column for both the "temp" and "cleaning" datasets. This allows us to compare the distribution of this variable between the two datasets and identify any differences or patterns.

Finally, we create a new dataset called "temp2" that only includes rows where the "annual_inc_merged" value is less than 100000. This helps us to further narrow down the data and focus on a specific subset of the data for analysis.

Overall, this process is useful because it helps us understand the characteristics and distribution of the data, identify any issues or abnormalities, and make informed decisions about how to proceed with our analysis. It is an important step in the data science process and ensures that our insights and conclusions are based on high-quality, accurate data.

In the next lines of code, we are working with a dataset called "cleaning" and manipulating a column called "issue_d." The first line uses the "substr" function to extract a specific portion of the "issue_d" values, namely the characters in positions 5 through 8.

The second line uses the "unique" function to identify the unique values of the modified "issue_d" column, which now only includes the characters extracted in the previous step.

The third line uses the “mutate” function and the “substr” function to replace the original “issue_d” column with the modified version that only includes the characters extracted earlier.

Next, we create a vector called “group1” that contains a list of values. We then use the “mutate” function and the “ifelse” function to create a new column called “year_group.” This column is populated with the value “Group1” if the “issue_d” value is included in the “group1” vector, or “Group2” otherwise. The “select” function is then used to remove the original “issue_d” column from the dataset.

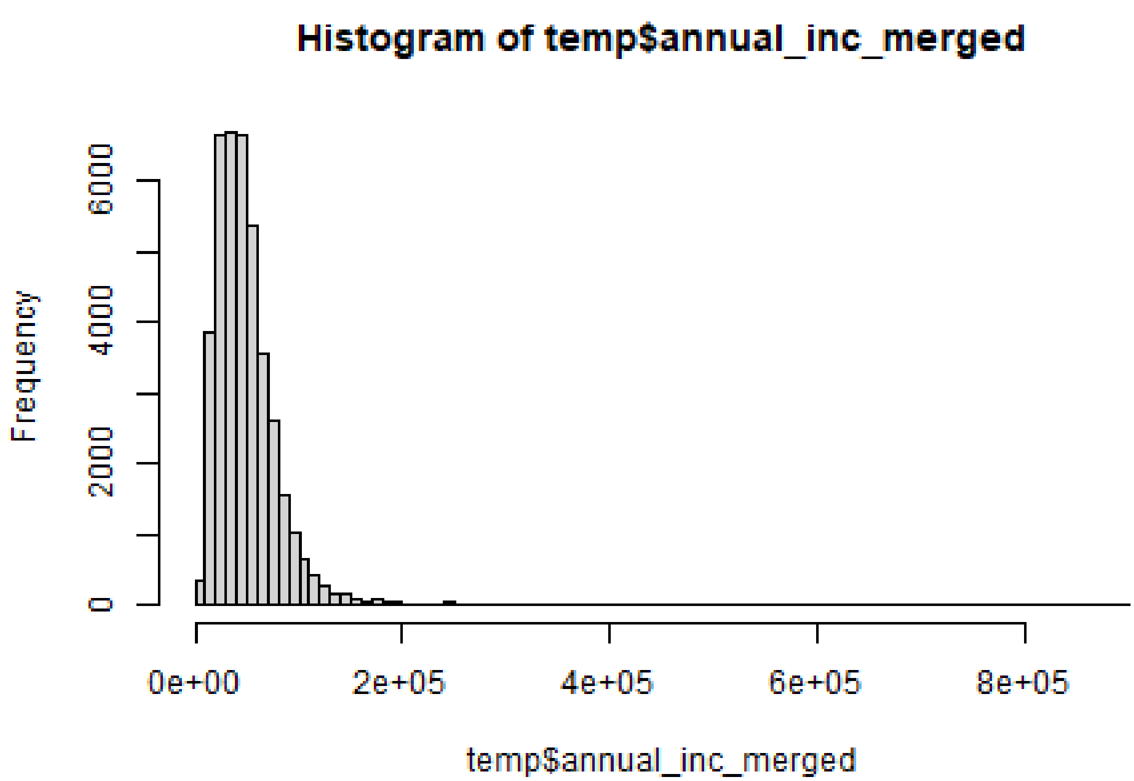
Finally, we use the “as.factor” function to convert the “year_group” column to a factor variable.

This process is useful for extracting and manipulating specific portions of the data, and for creating new variables based on the values of existing columns. It allows us to better understand the characteristics and patterns in the data and to conduct more targeted analyses.

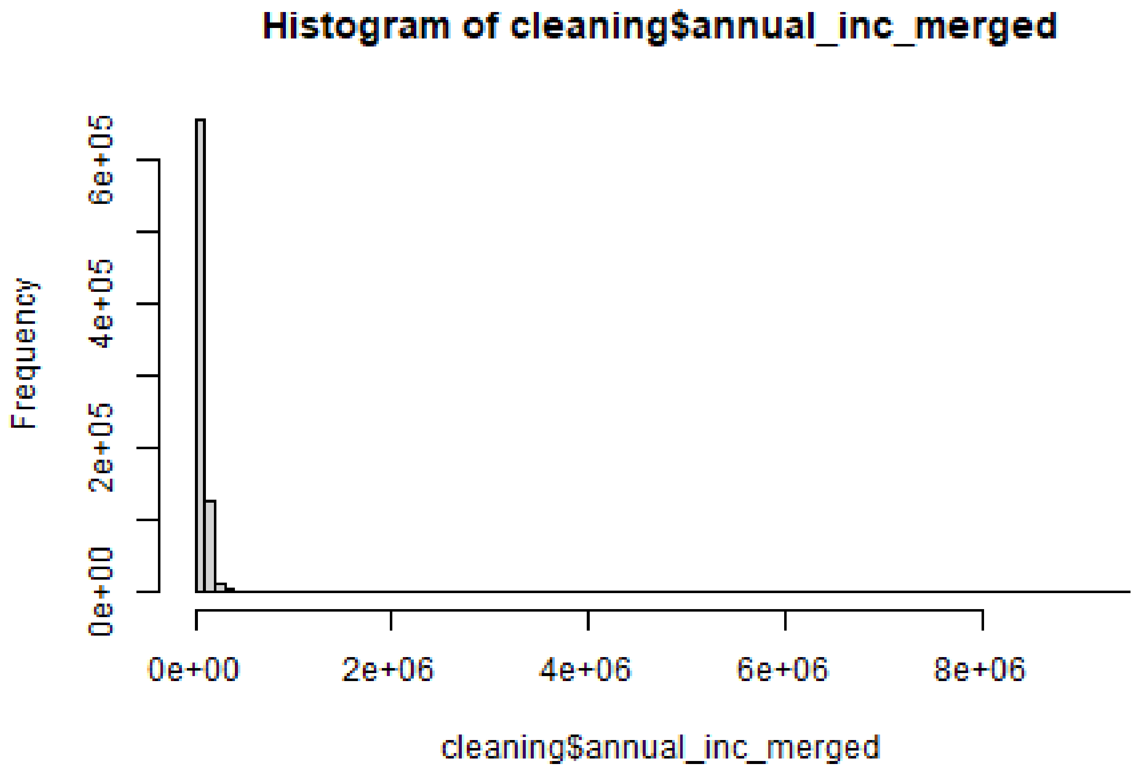
```r
unique(cleaning$emp_length)
```

```
##  [1] 1 year    10+ years 2 years   3 years   4 years   5 years   6 years
##  [8] < 1 year  9 years   n/a       7 years   8 years
## 12 Levels: < 1 year 1 year 10+ years 2 years 3 years 4 years ... n/a
```

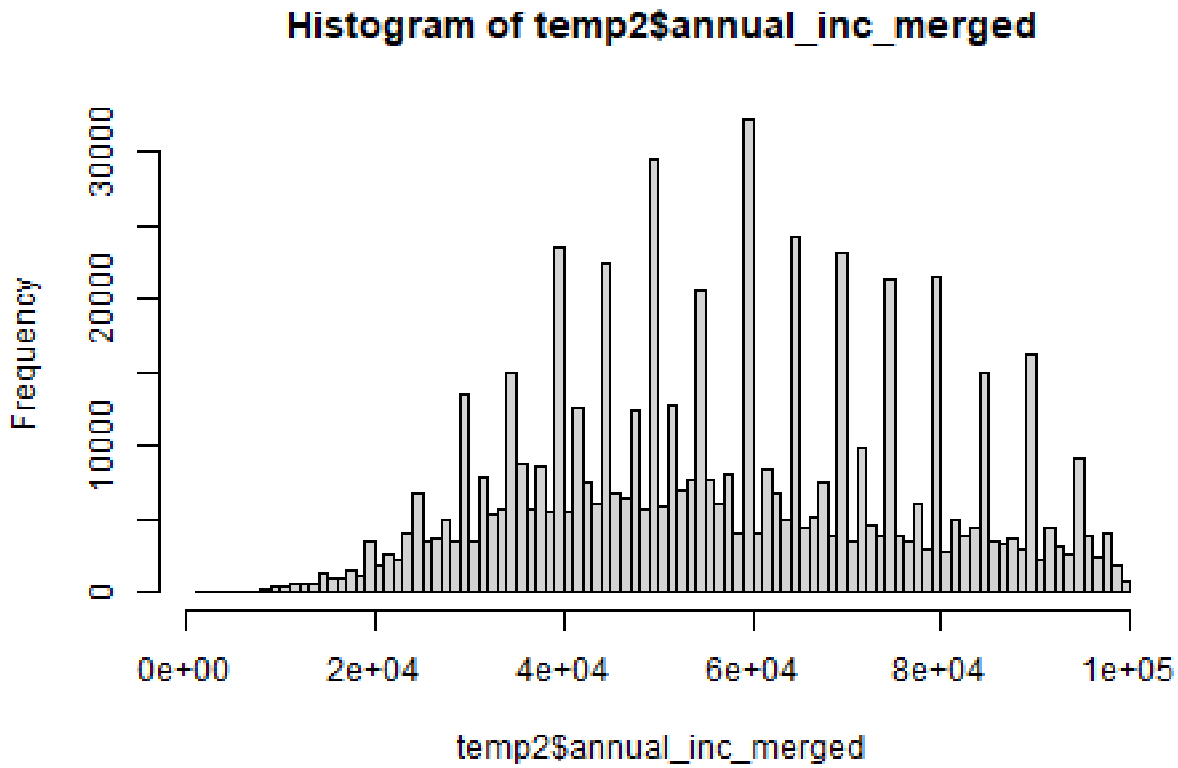
```r
temp<-cleaning %>% filter(emp_length=="n/a")
hist(temp$annual_inc_merged,breaks = 100)
```




```r
hist(cleaning$annual_inc_merged,breaks = 100)
```



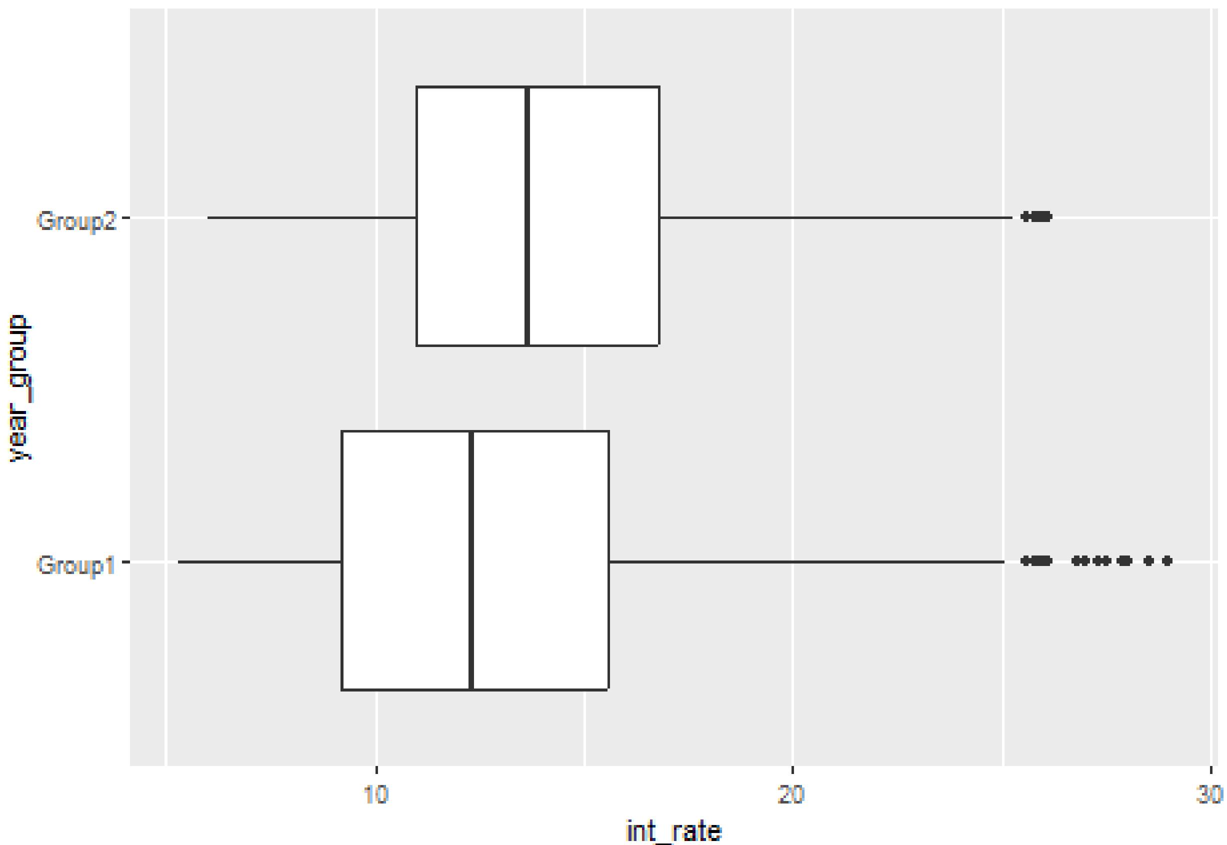

```
temp2<-cleaning %>% filter(annual_inc_merged<100000)
hist(temp2$annual_inc_merged,breaks = 100)
```




```r
unique(substr(cleaning$issue_d,5,8))
```

```
## [1] "2013" "2011" "2014" "2012" "2010" "2015" "2009" "2008" "2007"
```

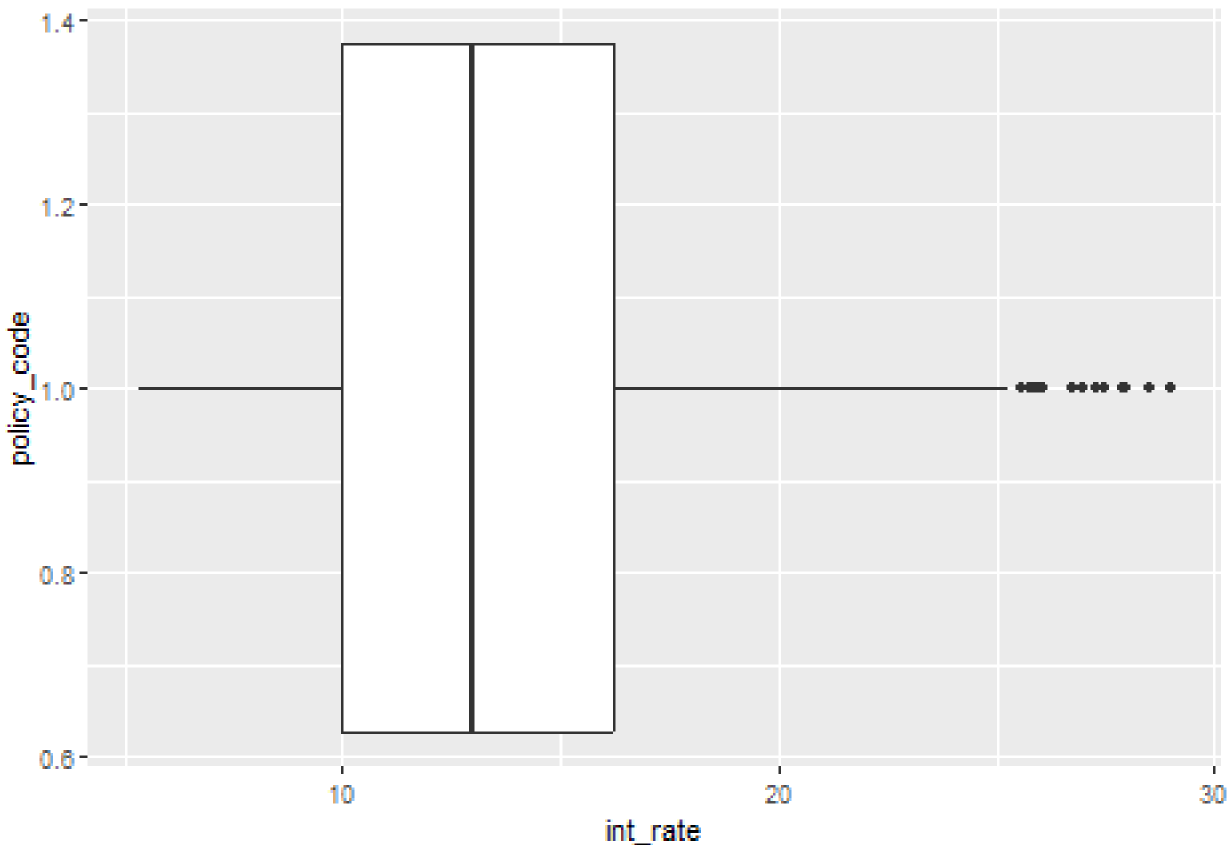
```r
cleaning <- cleaning %>% mutate(
    issue_d = substr(cleaning$issue_d,5,8))



group1 <- c("2007","2008","2010","2015","2011")
cleaning <- cleaning %>% mutate(
    year_group = ifelse(issue_d %in% group1,"Group1", "Group2")) %>% select(-issue_d)

 cleaning$year_group <- as.factor(cleaning$year_group)

ggplot(data = cleaning, mapping = aes(x=int_rate,y=year_group))+geom_boxplot()
```



Cleaning of plicy code There is only policy code 1, therefore delete

the column

```
ggplot(data = cleaning, mapping = aes(x=int_rate,y=policy_code))+geom_boxplot()
```

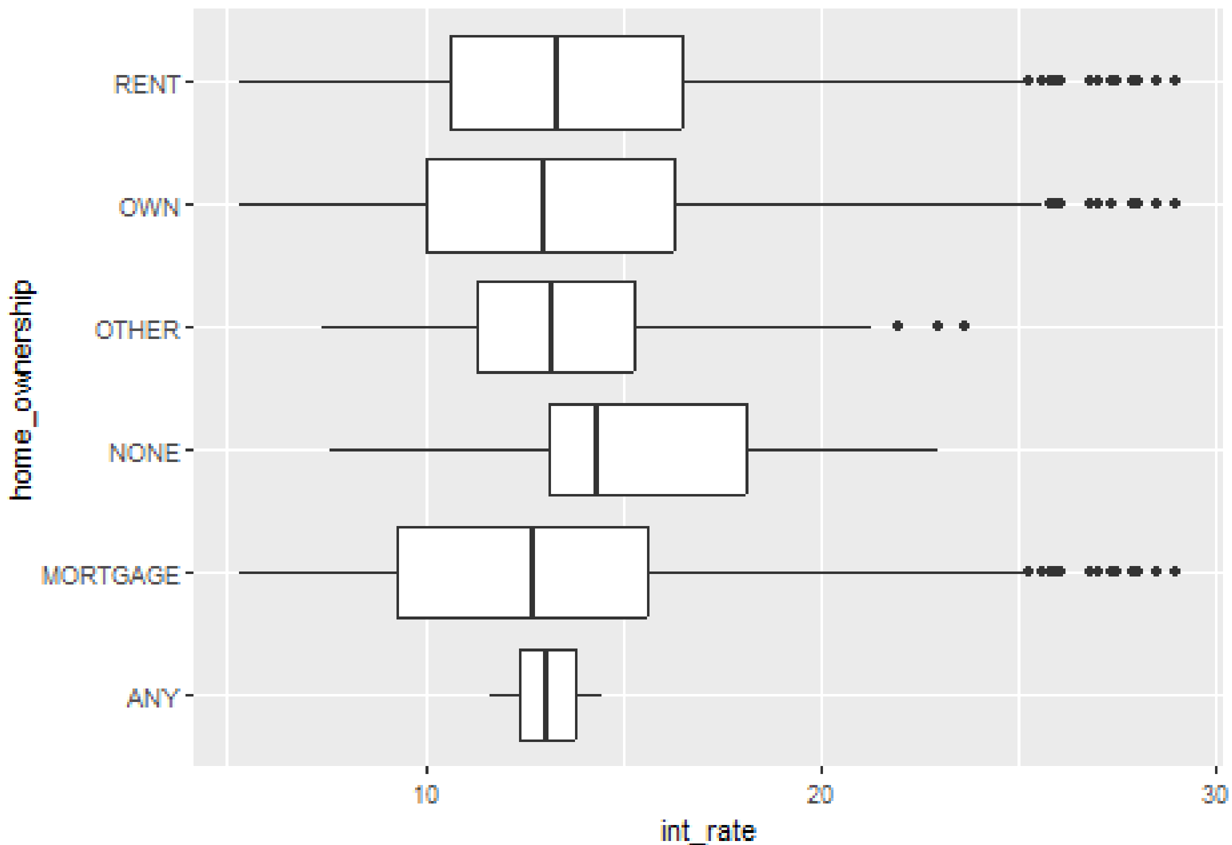



```
cleaning <- cleaning %>% select(-policy_code)
```

Cleaning of home_ownership During the data cleaning process, we

observed that the "home_ownership" variable does not appear to exhibit a clear correlation with interest rates. Specifically, the categories "ANY" and "OTHER" contain 2 and 154 cases, respectively, while the category "NONE" contains 39 cases. While the "NONE" category appears to have a higher interest rate than the other categories, the small sample size of 39 cases raises concerns about the validity of this observation. It is worth noting that the "NONE" category may potentially be associated with individuals who are homeless, which raises ethical considerations about granting loans to this population. Therefore, it is recommended to factorize the "home_ownership" column and rerun the analysis to ensure that deleted rows are not retained.

```
ggplot(data = cleaning, mapping = aes(x=int_rate,y=home_ownership))+geom_boxplot()
```




```
cleaning <- cleaning %>% filter(home_ownership %in% c("MORTGAGE","OWN","RENT") )
cleaning$home_ownership <- as.factor(cleaning$home_ownership)
```

Delete column zip code "The"character" column contains an excessive

number of unique values, making it difficult to accurately categorize the data. As a result, it is advisable to remove this column from the dataset to avoid introducing unnecessary complexity into the analysis."

```
cleaning <- cleaning %>% select(-zip_code)
```

Merge column addr_state

A common way of referring to regions in the United States is grouping them into 5 regions according to their geographic position on the continent: the Northeast:PA, NY, NJ, CT, RI, MA, VT, NH, ME, DE, MD Southwest:AZ, CA, CO, NV, NM, UT Northwest: ID, MT, OR, WA, WI, AK Southeast:AL, FL, GA, KY, MS, SC, NC, TN, VA, WV Midwest:IL, IN, IA, KS, MI, MN, MO, NE, ND, OH, SD, WI,
South:AR, LA, OK, TX

```
Northeast <- c("PA","NY","NJ","CT","RI","MA","VT","NH","ME","DE","MD")
Southwest <- c("AZ","CA","CO","NV","NM","UT")
Northwest <- c("ID","MT","OR","WA","WI","AK")
Southeast <- c("AL","FL","GA","KY","MS","SC","NC","TN","VA","WV")
Midwest <- c("IL","IN","IA","KS","MI","MN","MO","NE","ND","OH","SD","WI")
```

```
South <- c("AR","LA","OK","TX")

cleaning <- cleaning %>% mutate(
    region = ifelse(addr_state %in% Northeast,"northeast",
              ifelse(addr_state %in% Southwest,"southwest",
               ifelse(addr_state %in% Northwest,"northwest",
                 ifelse(addr_state %in% Southeast,"southeast",
                  ifelse(addr_state %in% Midwest,"midwest","south"))))))

  cleaning <- cleaning %>% select(-addr_state)
   cleaning$region <- as.factor(cleaning$region)
```

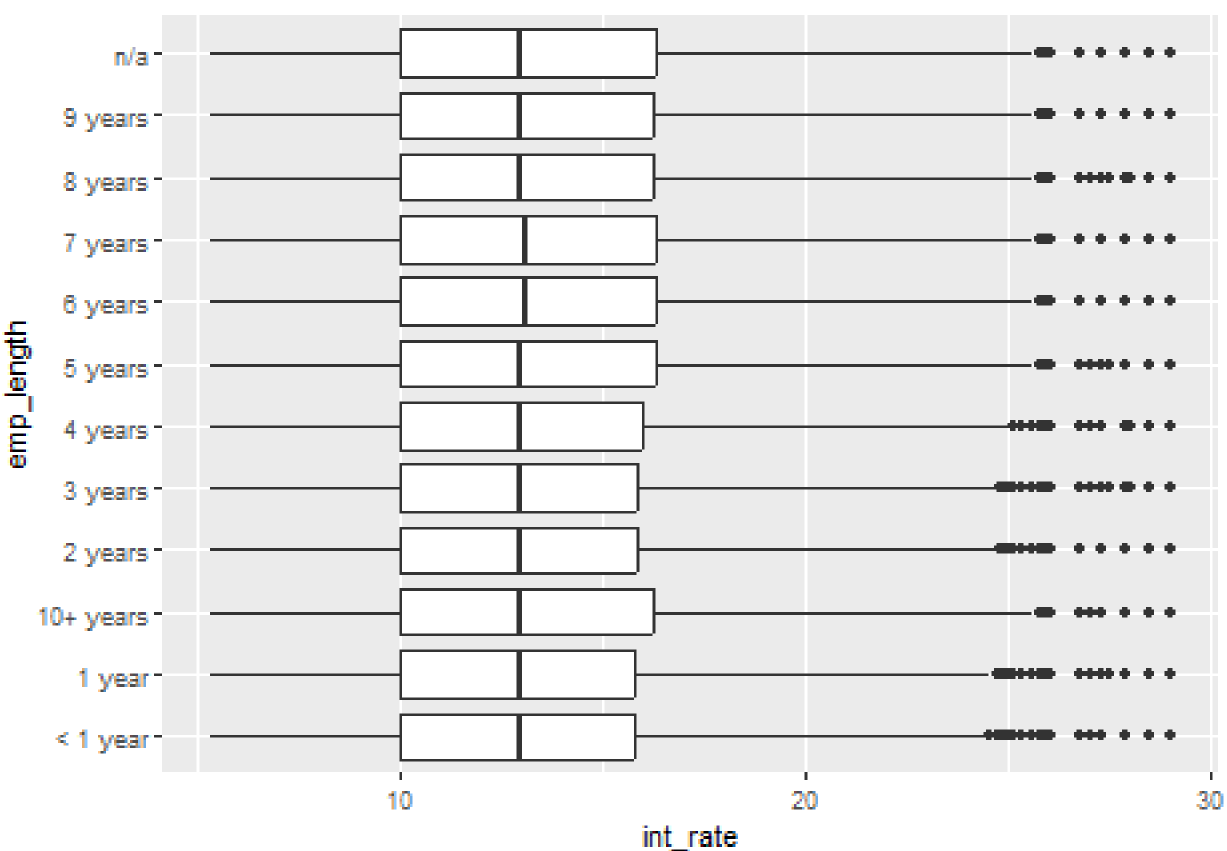
Cleaning of earliest_cr_line Last but not least just deleting

earliest_cr_line because that information is already covered through colums like inquieries, employed since and so on.

```
cleaning <- cleaning %>% select(-earliest_cr_line)
```

Cleaning of emp_length

```
ggplot(data = cleaning, mapping = aes(x=int_rate,y=emp_length))+geom_boxplot()
```



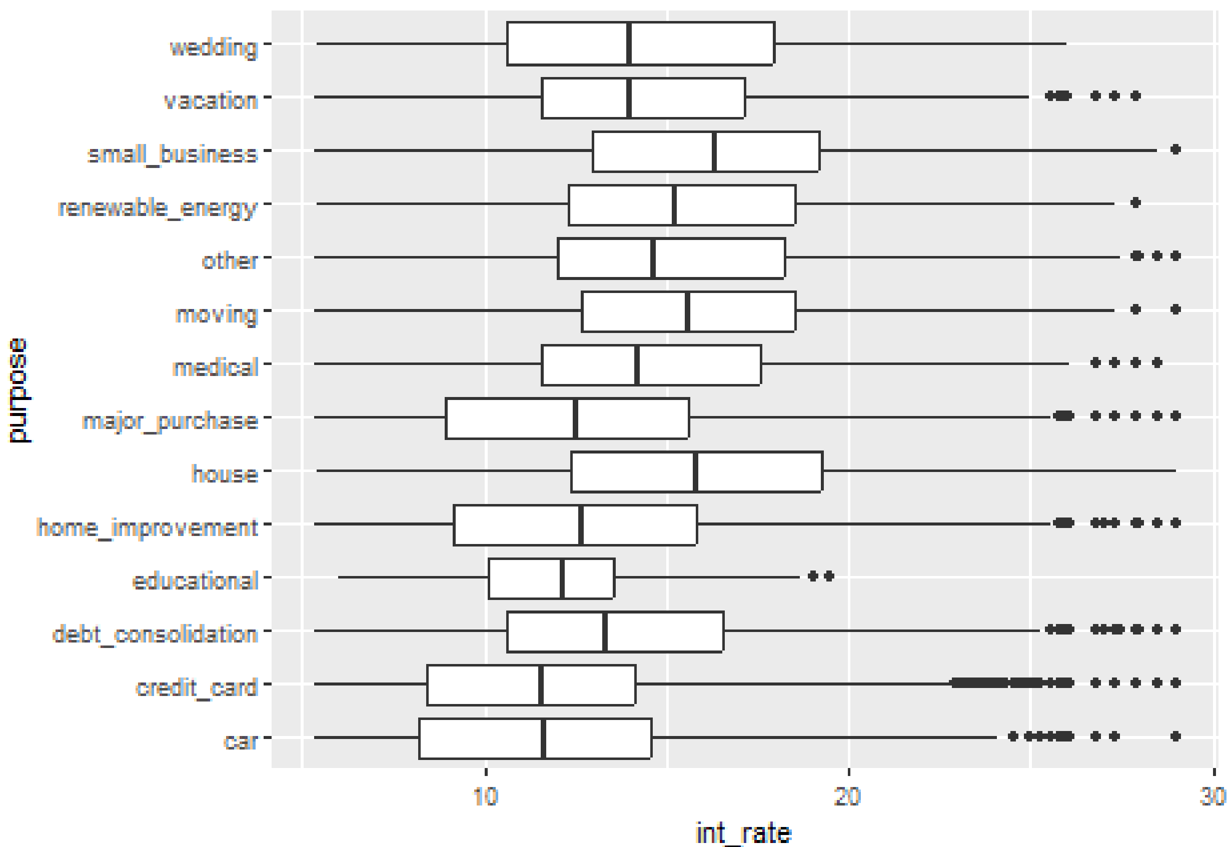
First we thought about cleaning emp_length because it still has 236966 n/a's. But after plotting emp_lenght it's clear that it does not have an impact on the interest. Therefore we delete this column.

```
cleaning <- cleaning %>% select(-emp_length)
```

Inspecting purpose Checking if other in purpose is really just other

or n/a. It is other! and the whole purpose is important for the interest, seen when plotting it.

```
ggplot(data = cleaning, mapping = aes(x=int_rate,y=purpose))+geom_boxplot()
```



After seeing the results, it looks like the purpose does have an impact on the interest

See results of the cleaning process

```
summary(cleaning)
```

```
##    loan_amnt            term           int_rate      installment
##  Min.   :  500   36 months:558413   Min.   : 5.32   Min.   :  15.67
##  1st Qu.: 8000   60 months:239478   1st Qu.: 9.99   1st Qu.: 260.71
##  Median :13000                      Median :12.99   Median : 382.55
```

```
##  Mean   :14758                      Mean   :13.24   Mean   : 436.74
##  3rd Qu.:20000                      3rd Qu.:16.20   3rd Qu.: 572.72
##  Max.   :35000                      Max.   :28.99   Max.   :1445.46
##
##   home_ownership       verification_status              purpose
##  MORTGAGE:398891   Not Verified   :240019   debt_consolidation:471654
##  OWN     : 78722   Source Verified:296478   credit_card       :185353
##  RENT    :320278   Verified       :261394   home_improvement  : 46459
##                                             other             : 38611
##                                             major_purchase    : 15549
##                                             small_business    :  9349
##                                             (Other)           : 30916
##   delinq_2yrs      inq_last_6mths       open_acc        pub_rec
##  Min.   : 0.0000   Min.   : 0.0000   Min.   : 1.00   Min.   : 0.0000
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 8.00   1st Qu.: 0.0000
##  Median : 0.0000   Median : 0.0000   Median :11.00   Median : 0.0000
##  Mean   : 0.3143   Mean   : 0.6945   Mean   :11.55   Mean   : 0.1954
##  3rd Qu.: 0.0000   3rd Qu.: 1.0000   3rd Qu.:14.00   3rd Qu.: 0.0000
##  Max.   :39.0000   Max.   :33.0000   Max.   :90.00   Max.   :63.0000
##
##    revol_bal         revol_util       total_acc      initial_list_status
##  Min.   :      0   Min.   :  0.00   Min.   :  1.00   f:410580
##  1st Qu.:   6451   1st Qu.: 37.70   1st Qu.: 17.00   w:387311
##  Median :  11882   Median : 56.00   Median : 24.00
##  Mean   :  16934   Mean   : 55.05   Mean   : 25.27
##  3rd Qu.:  20844   3rd Qu.: 73.50   3rd Qu.: 32.00
##  Max.   :2904836   Max.   :892.30   Max.   :169.00
##
##    out_prncp     out_prncp_inv   collections_12_mths_ex_med acc_now_delinq
##  Min.   :    0   Min.   :    0   Min.   : 0.00000           Min.   : 0.000000
##  1st Qu.:    0   1st Qu.:    0   1st Qu.: 0.00000           1st Qu.: 0.000000
##  Median : 6465   Median : 6460   Median : 0.00000           Median : 0.000000
##  Mean   : 8407   Mean   : 8403   Mean   : 0.01448           Mean   : 0.005026
##  3rd Qu.:13664   3rd Qu.:13660   3rd Qu.: 0.00000           3rd Qu.: 0.000000
##  Max.   :49373   Max.   :49373   Max.   :20.00000           Max.   :14.000000
##
##   tot_coll_amt      tot_cur_bal        open_acc_6m        open_il_6m
##  Min.   :      0   Min.   :      0   Min.   : 0.00000   Min.   : 0.00000
##  1st Qu.:      0   1st Qu.:  23206   1st Qu.: 0.00000   1st Qu.: 0.00000
##  Median :      0   Median :  65420   Median : 0.00000   Median : 0.00000
##  Mean   :    210   Mean   : 128477   Mean   : 0.02641   Mean   : 0.06983
##  3rd Qu.:      0   3rd Qu.: 195890   3rd Qu.: 0.00000   3rd Qu.: 0.00000
##  Max.   :9152545   Max.   :8000078   Max.   :14.00000   Max.   :33.00000
##
##   open_il_12m        open_il_24m       mths_since_rcnt_il  total_bal_il
##  Min.   : 0.00000   Min.   : 0.00000   Min.   :  0.0000   Min.   :     0.0
##  1st Qu.: 0.00000   1st Qu.: 0.00000   1st Qu.:  0.0000   1st Qu.:     0.0
##  Median : 0.00000   Median : 0.00000   Median :  0.0000   Median :     0.0
##  Mean   : 0.01817   Mean   : 0.03992   Mean   :  0.4919   Mean   :   872.1
##  3rd Qu.: 0.00000   3rd Qu.: 0.00000   3rd Qu.:  0.0000   3rd Qu.:     0.0
##  Max.   :12.00000   Max.   :19.00000   Max.   :363.0000   Max.   :878459.0
##
##     il_util        open_rv_12m        open_rv_24m         max_bal_bc
##  Min.   :  0.00   Min.   : 0.00000   Min.   : 0.00000   Min.   :    0.0
```

```
##  1st Qu.:  0.00   1st Qu.: 0.00000   1st Qu.: 0.00000   1st Qu.:    0.0
##  Median :  0.00   Median : 0.00000   Median : 0.00000   Median :    0.0
##  Mean   :  1.49   Mean   : 0.03316   Mean   : 0.07115   Mean   :  140.8
##  3rd Qu.:  0.00   3rd Qu.: 0.00000   3rd Qu.: 0.00000   3rd Qu.:    0.0
##  Max.   :223.30   Max.   :22.00000   Max.   :43.00000   Max.   :83047.0
##
##     all_util       total_rev_hi_lim      inq_fi          total_cu_tl
##  Min.   :  0.000   Min.   :      0   Min.   : 0.00000   Min.   : 0.00000
##  1st Qu.:  0.000   1st Qu.:  11700   1st Qu.: 0.00000   1st Qu.: 0.00000
##  Median :  0.000   Median :  21800   Median : 0.00000   Median : 0.00000
##  Mean   :  1.457   Mean   :  29568   Mean   : 0.02262   Mean   : 0.03669
##  3rd Qu.:  0.000   3rd Qu.:  37900   3rd Qu.: 0.00000   3rd Qu.: 0.00000
##  Max.   :151.400   Max.   :9999999   Max.   :16.00000   Max.   :35.00000
##
##   inq_last_12m           mths_since_delinq_cat     mths_since_last_record_cat
##  Min.   :-4.00000   1_to_3_years     :150675    1_to_3_years     : 11811
##  1st Qu.: 0.00000   3_to_5_years     :100941    3_to_5_years     : 30524
##  Median : 0.00000   more_than_5_years: 61595    more_than_5_years: 77818
##  Mean   : 0.04734   No_delinq        :408518    No_record        :675618
##  3rd Qu.: 0.00000   recent           : 76162    recent           :  2120
##  Max.   :32.00000
##
##   mths_since_last_major_derog_cat annual_inc_merged   dti_merged
##  1_to_3_years     : 62170         Min.   :   1896   Min.   : 0.00
##  3_to_5_years     : 69157         1st Qu.:  45000   1st Qu.:11.91
##  more_than_5_years: 52327         Median :  65000   Median :17.66
##  No_derog         :598524         Mean   :  75037   Mean   :18.13
##  recent           : 15713         3rd Qu.:  90000   3rd Qu.:23.94
##                                   Max.   :9500000   Max.   :43.86
##
##   year_group          region
##  Group1:412079   midwest  :128925
##  Group2:385812   northeast:186148
##                  northwest: 41985
##                  south    : 94776
##                  southeast:173072
##                  southwest:172985
##
```

The data is cleaned now and can be used for the next part which is modeling

So for the modeling part, we want to check first each column with the cleaned dataset. Especially how they correlate to the interest rate which we want to predict. The findings here can be taken into account when building a model. But we have to keep in mind that it is possible for there to be a correlation between two columns in a linear regression model but no significance in xgboost. This can happen if the relationship between the two columns is non-linear and xgboost is better able to model non-linear relationships than linear regression. And vize versa.

```
#Save final file for the use in the regression model
cleaning <- data.frame(cleaning %>% dplyr::select(int_rate,loan_amnt, term,installment,home_ownership, 
saveRDS(cleaning, "../Data/Out/cleanData.rds")
```

Conclusion In the context of crowdlending, it is important to practice

diversification in order to mitigate risk. For example, if you have 20,000 to invest and you put it all into two projects, and one of them defaults (meaning it is more than 120 days late on payments), you will suffer significant losses. However, if you distribute your 20,000 across 200 projects and one of them defaults, your losses will be minimized. The overall default rate at Lending Club, a crowdlending platform, is 7%, which is significantly higher than the default rate of approximately 2% in the Swiss market. If you have a large number of retail investors (individuals with limited funds for investment) who are unable to diversify their investments, there is a higher risk of negative reputation due to the higher likelihood of retail investors incurring losses. As a result, Lending Club has recently stopped accepting retail investors and only allows institutional investors, who have sufficient funds to diversify their investments and minimize their losses. The high default rate of 7% at Lending Club suggests that the platform may have been lending money to a wide range of borrowers in order to achieve growth. The net annualized return (NAR), which takes into account the default rate, is 8.28%. Upon analyzing the plots of the cleaned dataset, it is apparent that Lending Club did not adequately consider a large portion of the available data in determining interest rates. In particular, individuals who may not have been creditworthy were still granted loans. This suggests that Lending Club's underwriting process may have been insufficient, leading to the decision to only accept institutional investors. This is likely an attempt to mitigate risk and improve the quality of their loans by relying on investors with the resources to perform more thorough evaluations of potential borrowers