Data Cleaning LCdata

2022-11-16

#Assigment 1: Data Cleaning

```
getwd()
```

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

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

remove NA's. Remove whole lines or replace them with a value. with the code which() and choosing the corect column, you can add TRUE so it will show you the row in that column where the NA is located. library(dplyr) #is the most usefull library, google it bitch!#calling the dplyr library by pressing alt+shift+m

in the data cleaning, I choose to use dplyr library to select the row anual income I save in the data cleaning the following; by calling filter is.na i say delete me all the rows that have na data in anual income, because as seen in the sumary, it's only 4 rows and therefore legit. By calling select i delete whole colums. With the minus in front of the name of the colums I say R which column to delete. mutate() adds new variables and preserves existing ones. the new colums is called _cat. Ifelse transforms months sice delinq into _cat. With the if else function I look at the month since last delinwuency. There were 400k NA's and the values were a tring. So the first task is to group that data by inspecting it in a histogram. We saw there that the data goes up to 500 months. The grouping is more subjective with business knowledge. After the grouping done by ifelse, they have to be tuned into numbers via

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

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

```
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
```

```
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)
```

So this now was the very basis. All the colums rows are deleted, the first NA's are cleaned and the mutation from a huge number of dataset into subsets as strings and then into numbers is done. This is the basis for the rest of the data cleaning. Starting with the NA's. delinq_2_years has 21 NA's. It's worth's go from 0 to 39. The column tells us how many "bad entries" did someone have in a certain register. So the first question to ask here is, how to replace the NA's? Delete the entire row? Or delete the entire column? So most of the cases had no delinq within the last two years. So what's the impact on the interest of the few ones that had a delinq?

revol_bal has only 2 NA's which means deleting the row! revol_util has 429 NA's which is more but compared to the dataset of 800k entries vernachlässigbar collections_12_mths_ex_med has 101 NA's which is more but compared to the dataset of 800k entries vernachlässigbar

#Summary of NA's

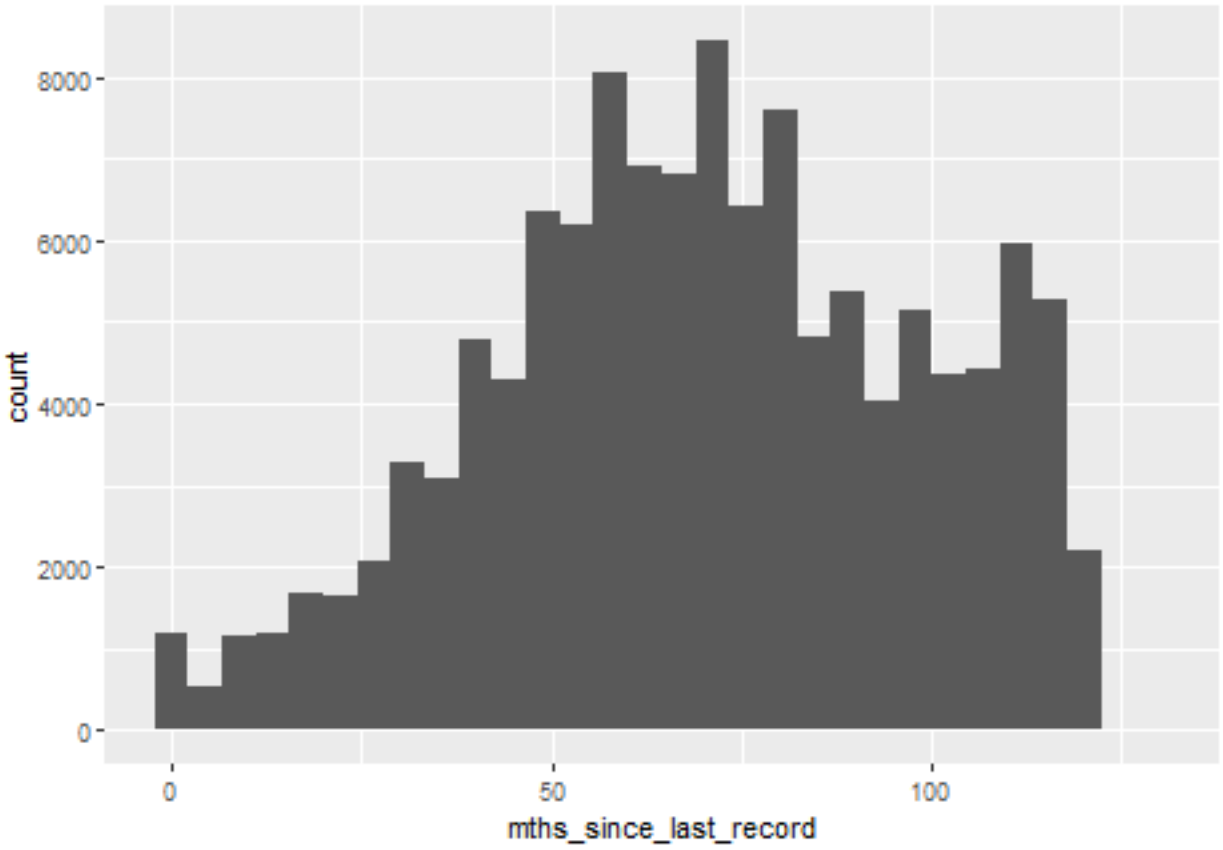
Now to the cases that have more than 1k NA's which should not be deleted, are the following:

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

```
library(ggplot2)
ggplot(data = cleaning, mapping = aes(x=mths_since_last_record))+geom_histogram()
```

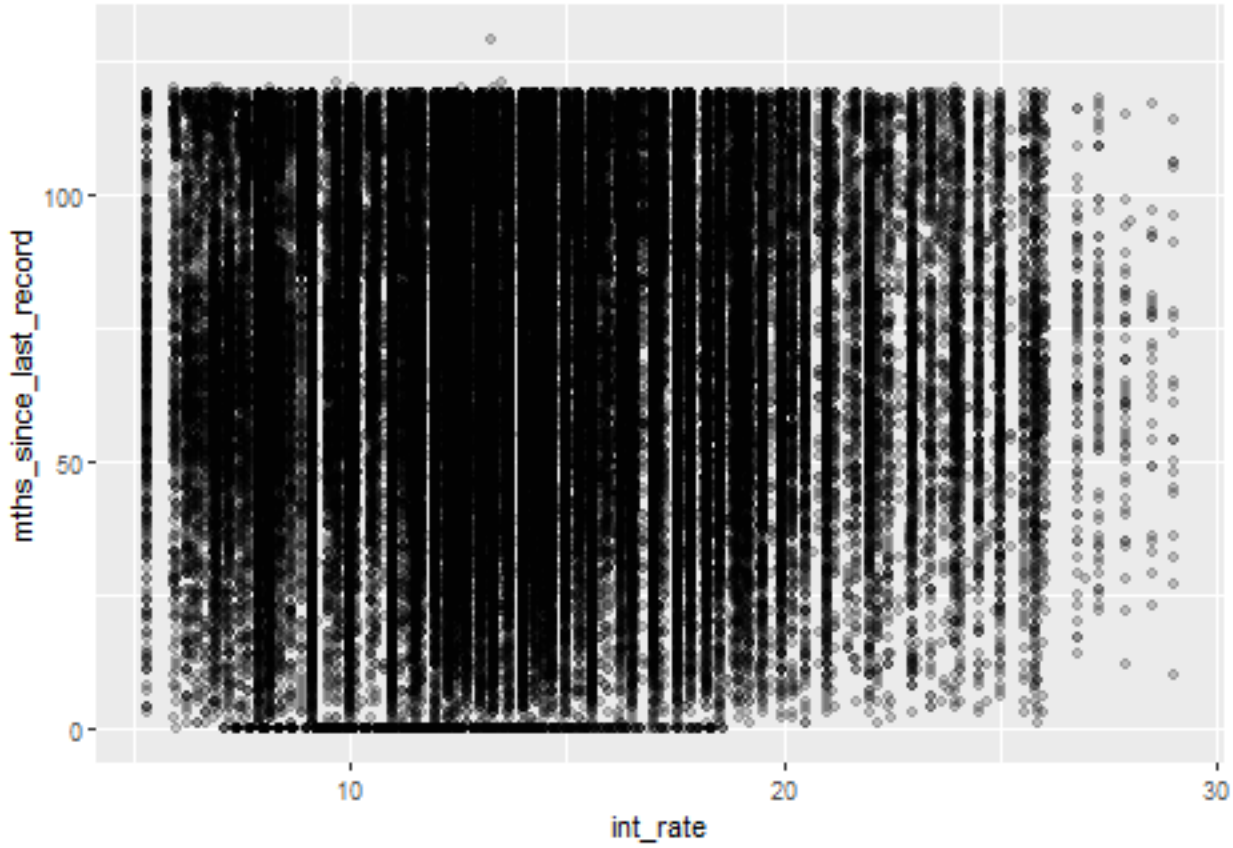
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## 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)
```

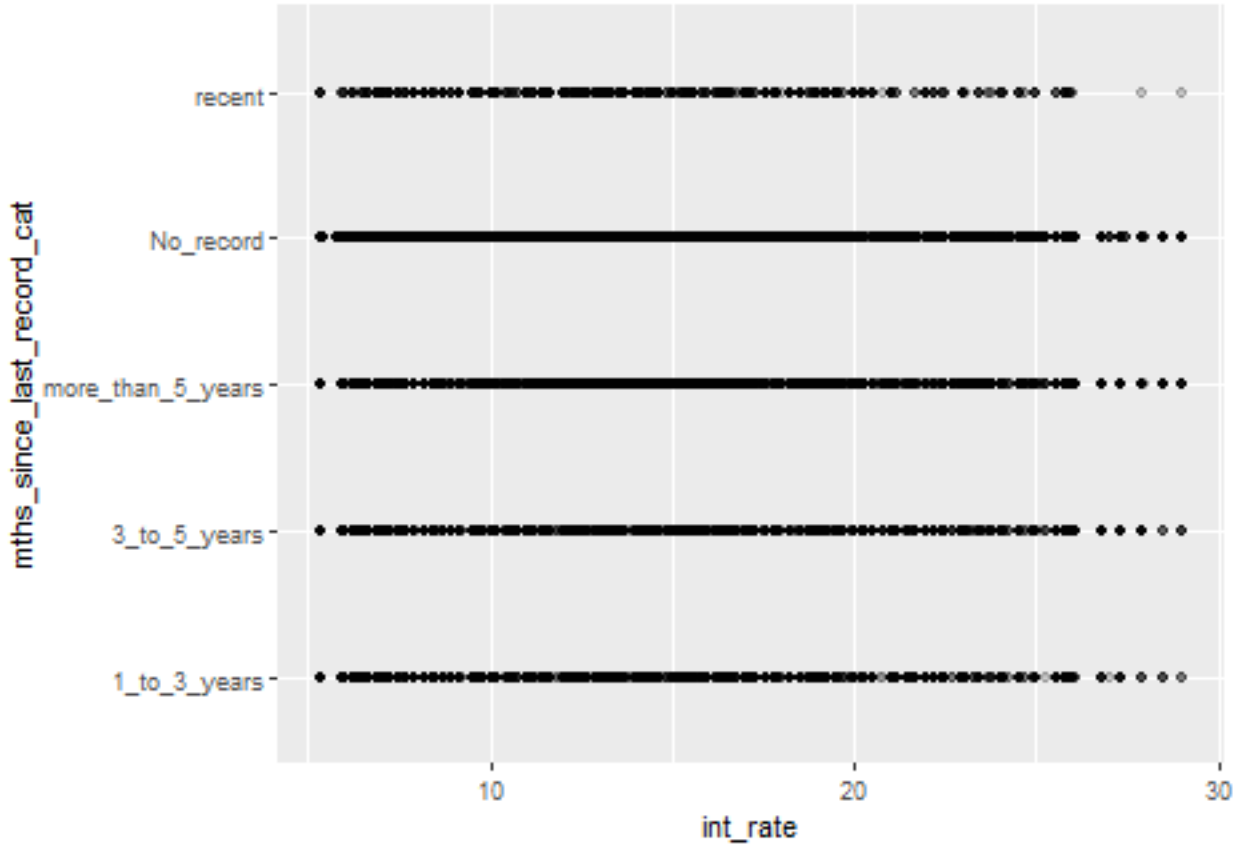
```
## Warning: Removed 674745 rows containing missing values (`geom_point()`).
```



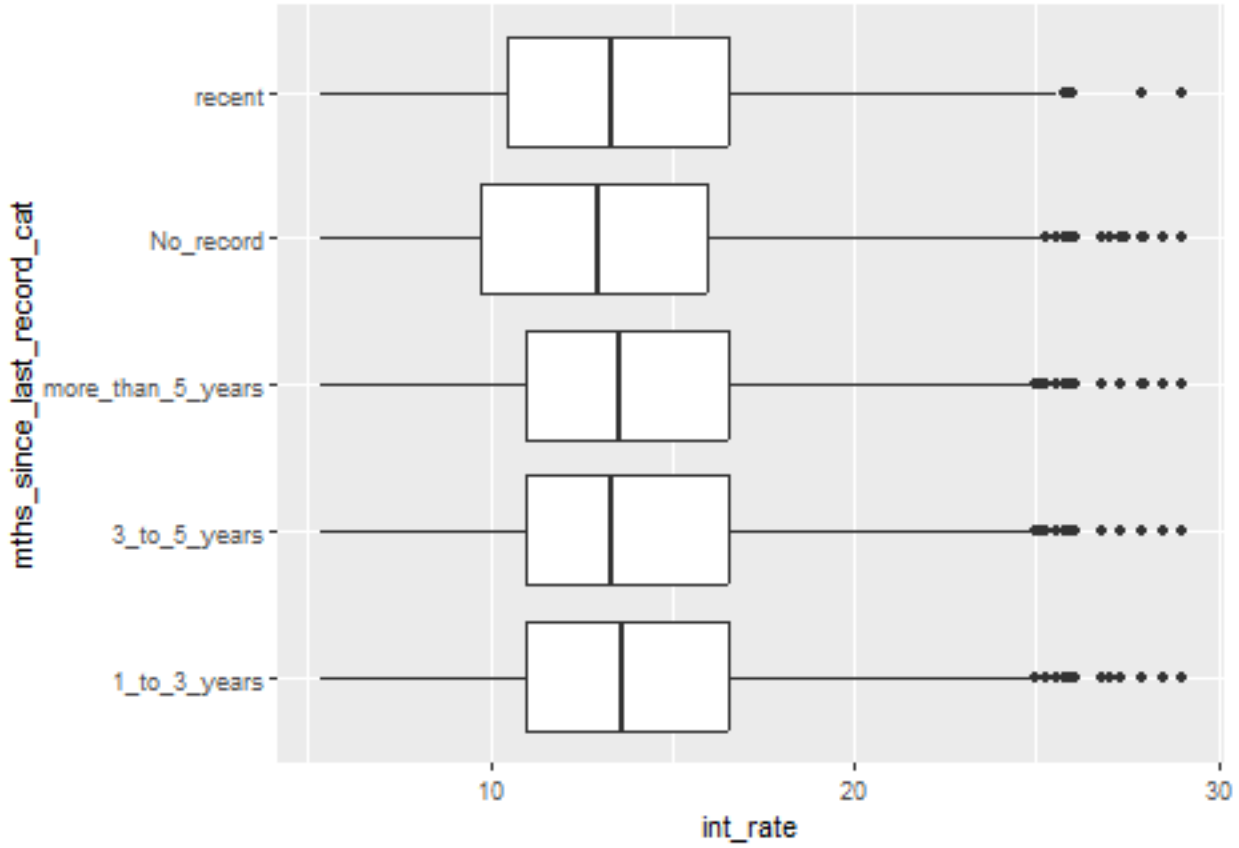
After plotting, no corelation could be detected. Therefore categorizing would be another try.

#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()
```

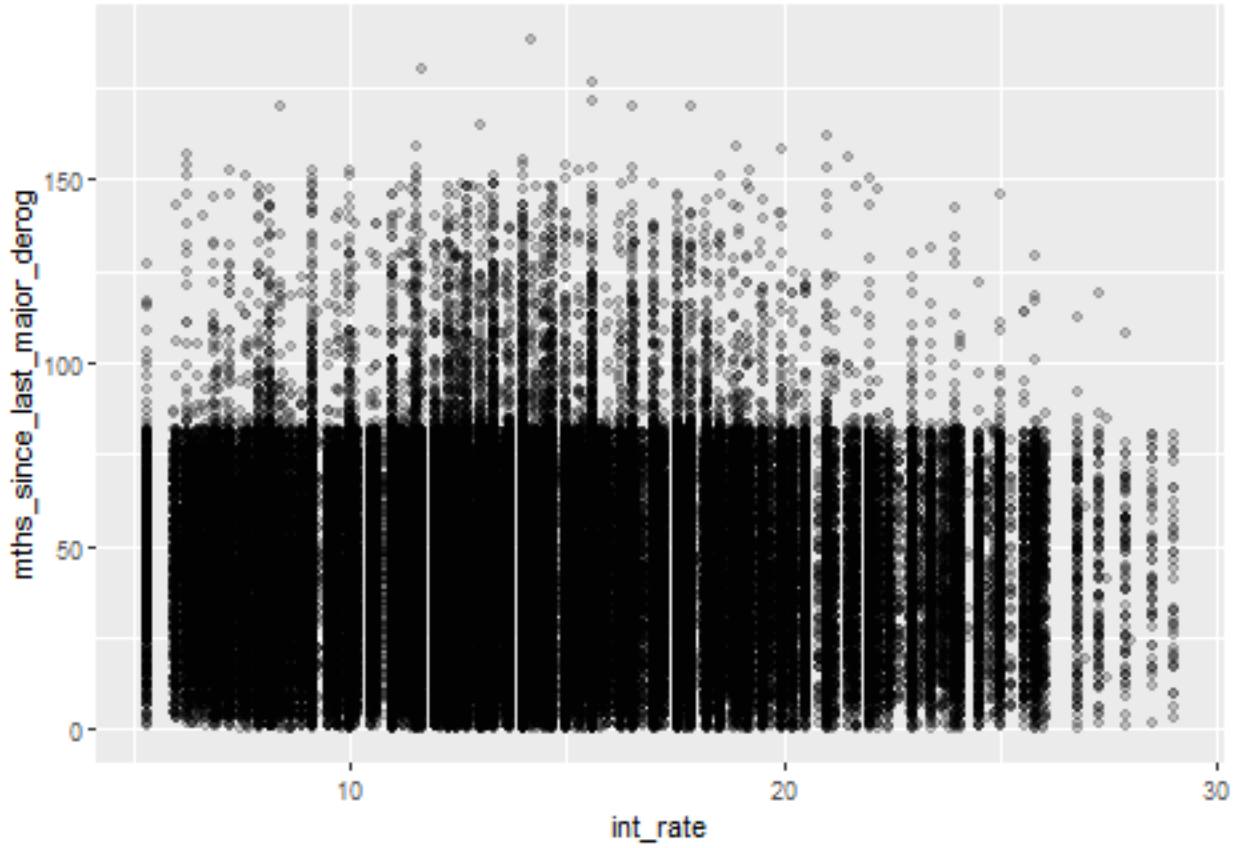


The results seem very odd. There is only a small, insignificant change in the interest rate when someone never had an entry in a public register. It doesn't seem to matter, if you just had a record compared to when you had a bad entry more then 5 years ago. It even seems to be better if you just had something negative which we believe, is not taken into account by Lending Club or they are doing a bad job in underwriting which may explain that they are out of business.

#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)
```

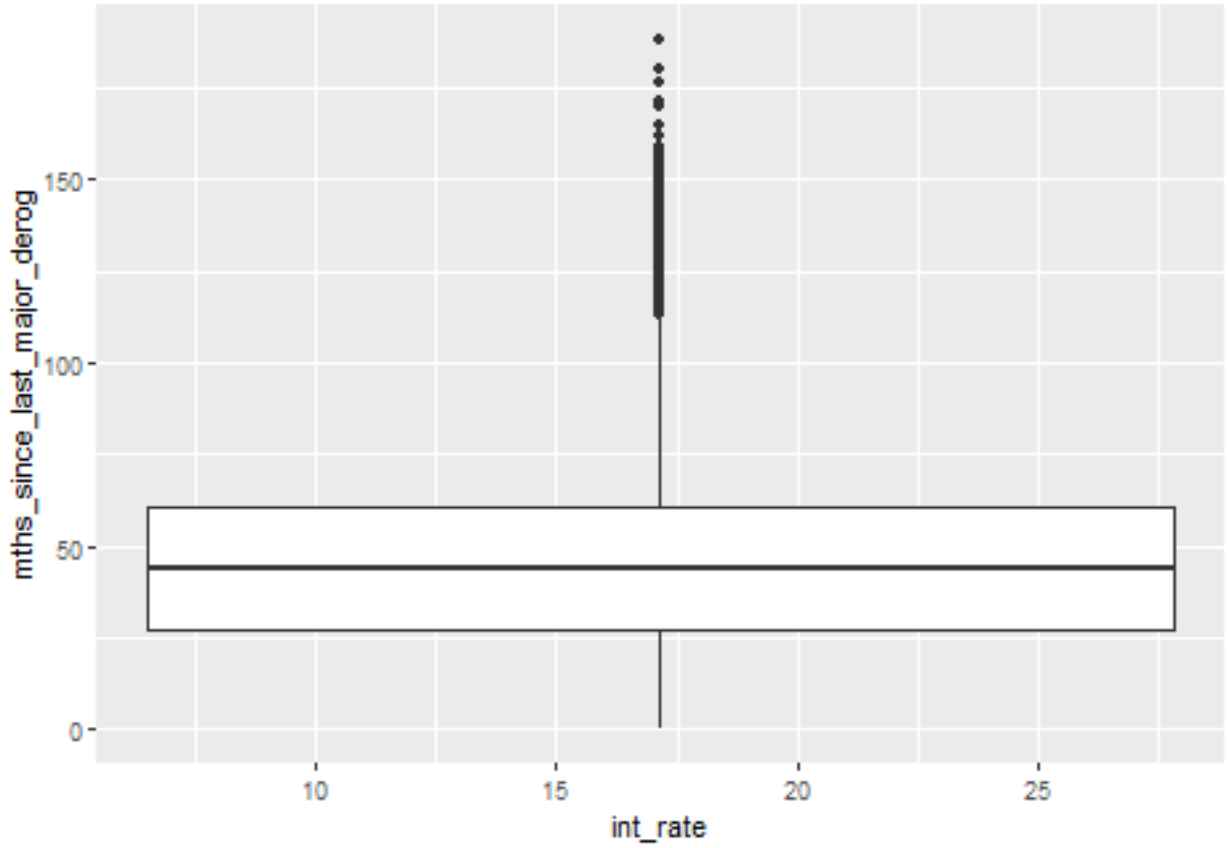
```
## Warning: Removed 598693 rows containing missing values (`geom_point()`).
```



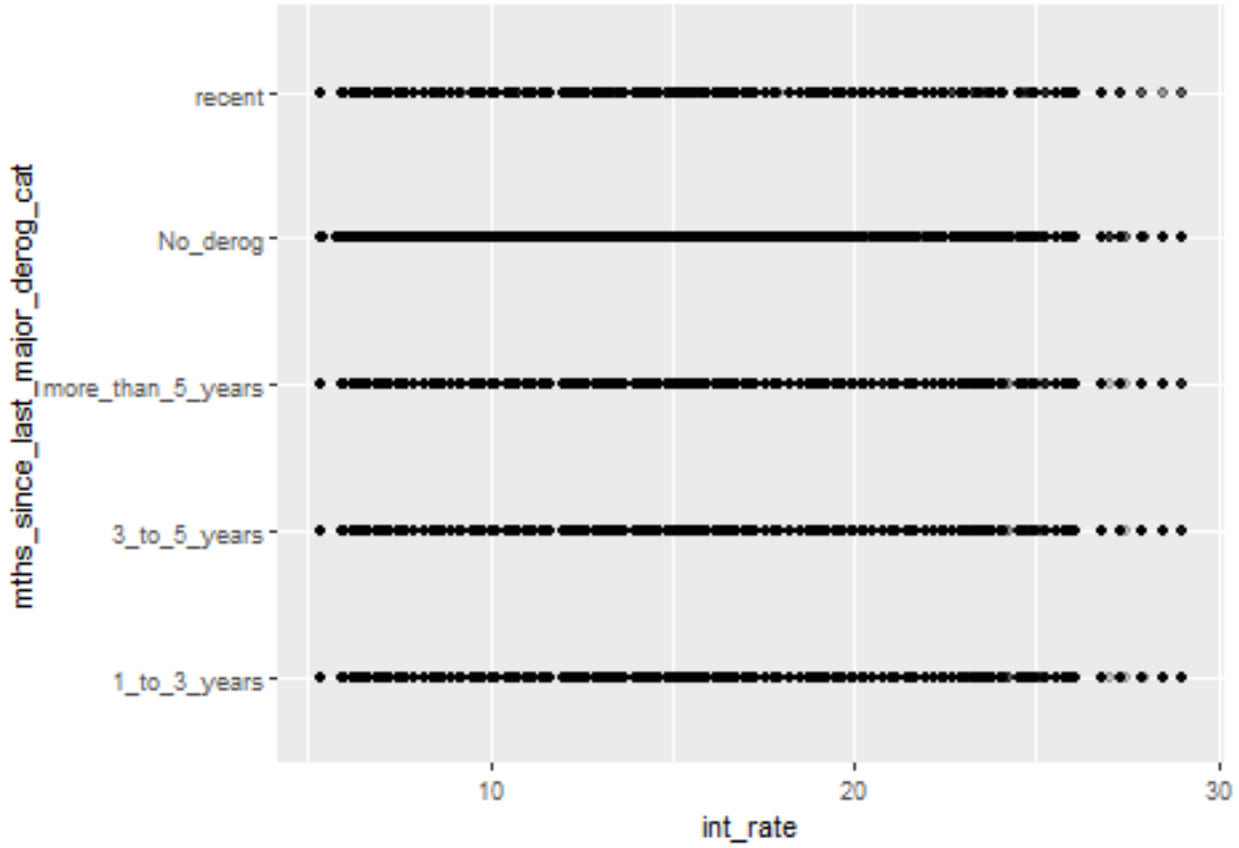

```
ggplot(data = cleaning, mapping = aes(x=int_rate,y=mths_since_last_major_derog))+geom_boxplot()
```

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

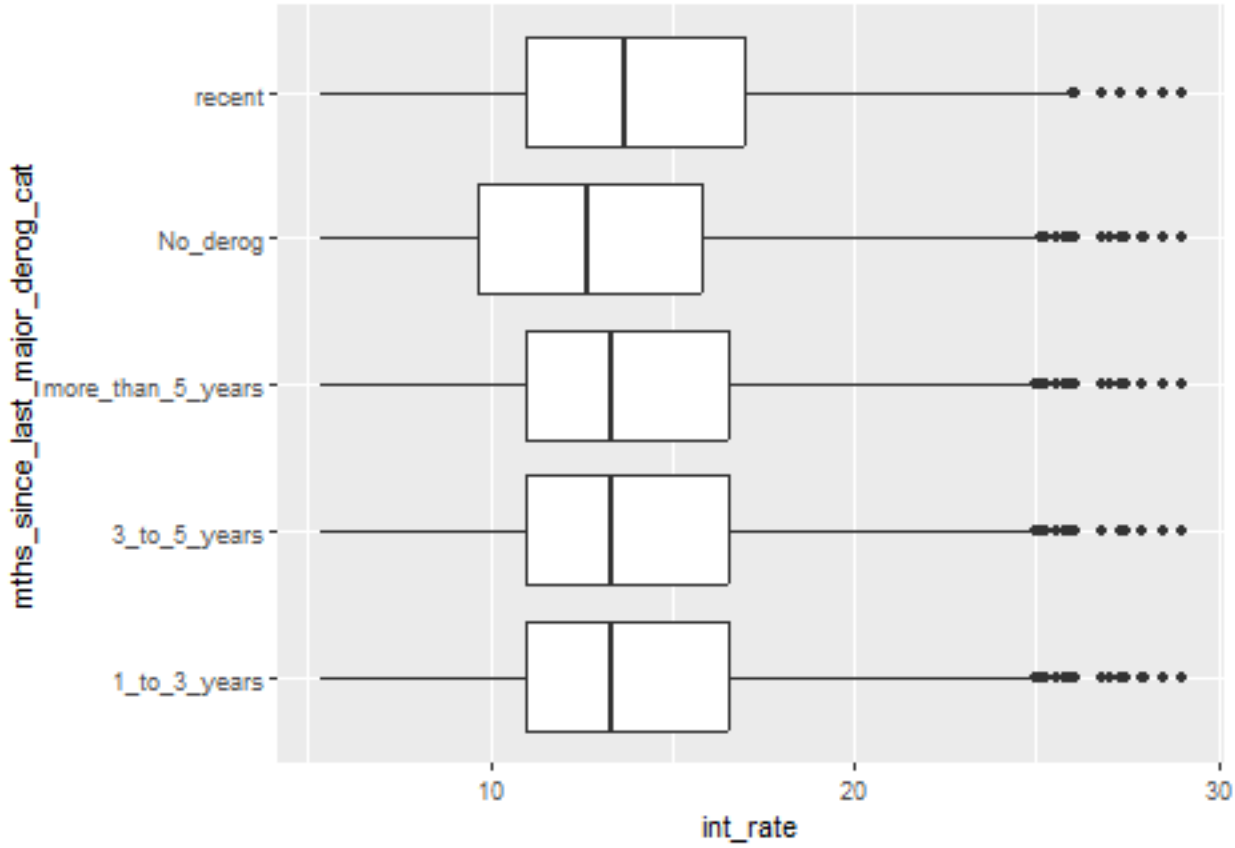
```
## Warning: Removed 598693 rows containing non-finite values (`stat_boxplot()`).
```




```
#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()
```



Same results for derog like for last record.... the data seems again very odd and a picture of how bad they might have done their underwriting forms.

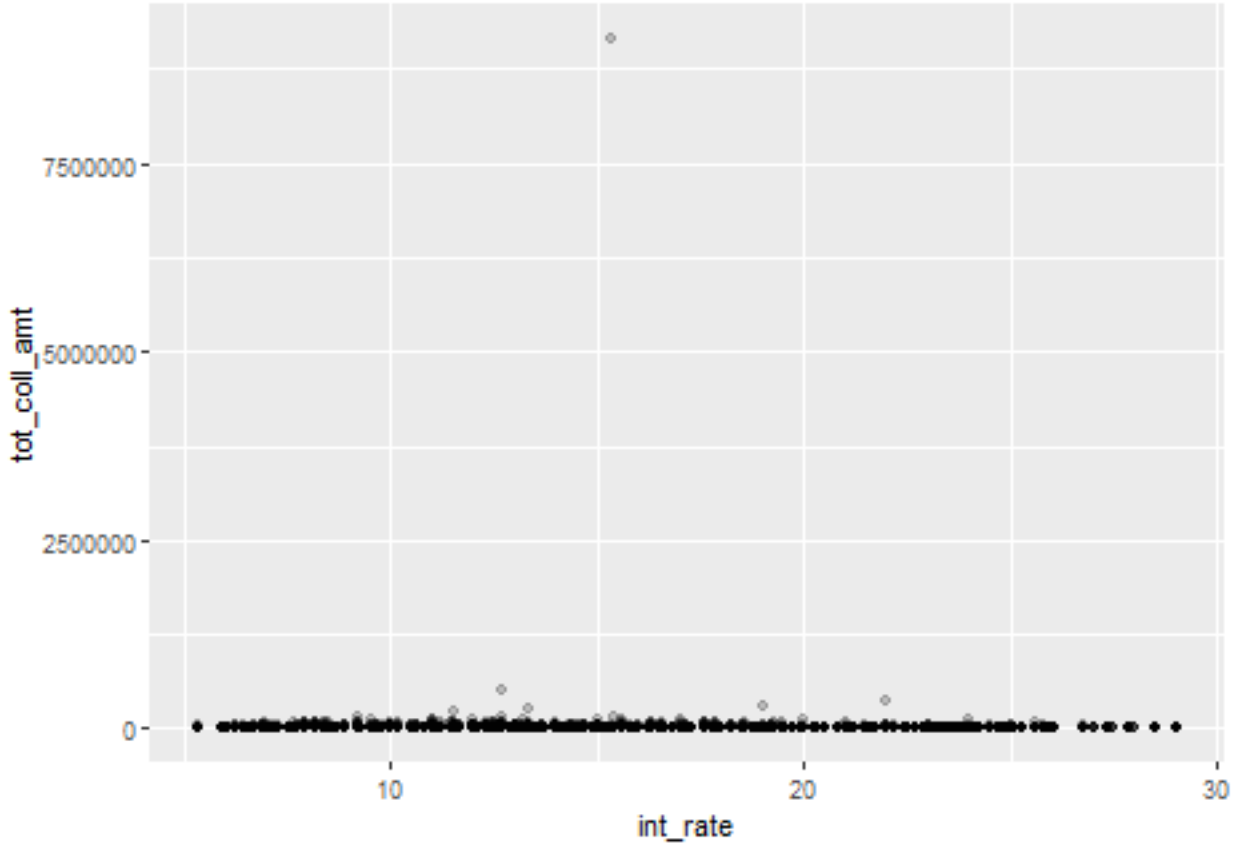
#Cleaning of annual_inc_joint and dti_joint Before cleaning, this data only shows, if there is a joint income. Same applies for dti and dti_joint. Therefore dti and annual income should be merged. So if somebody has a joint application, then joint income and joint dti should be taken. If somebody has an individual application, the values from annual_inc and dti should be taken. This is the correct way to clean it because some entries have the value zero for annual inc but a higher value for the joint inc which means that only the second applicant has an income. Same for dti. annual_inc_joint annual_inc mutate(address = ifelse(address == '',work_address,address))

```
#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)


#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 #ask SRI how to clean this outlier

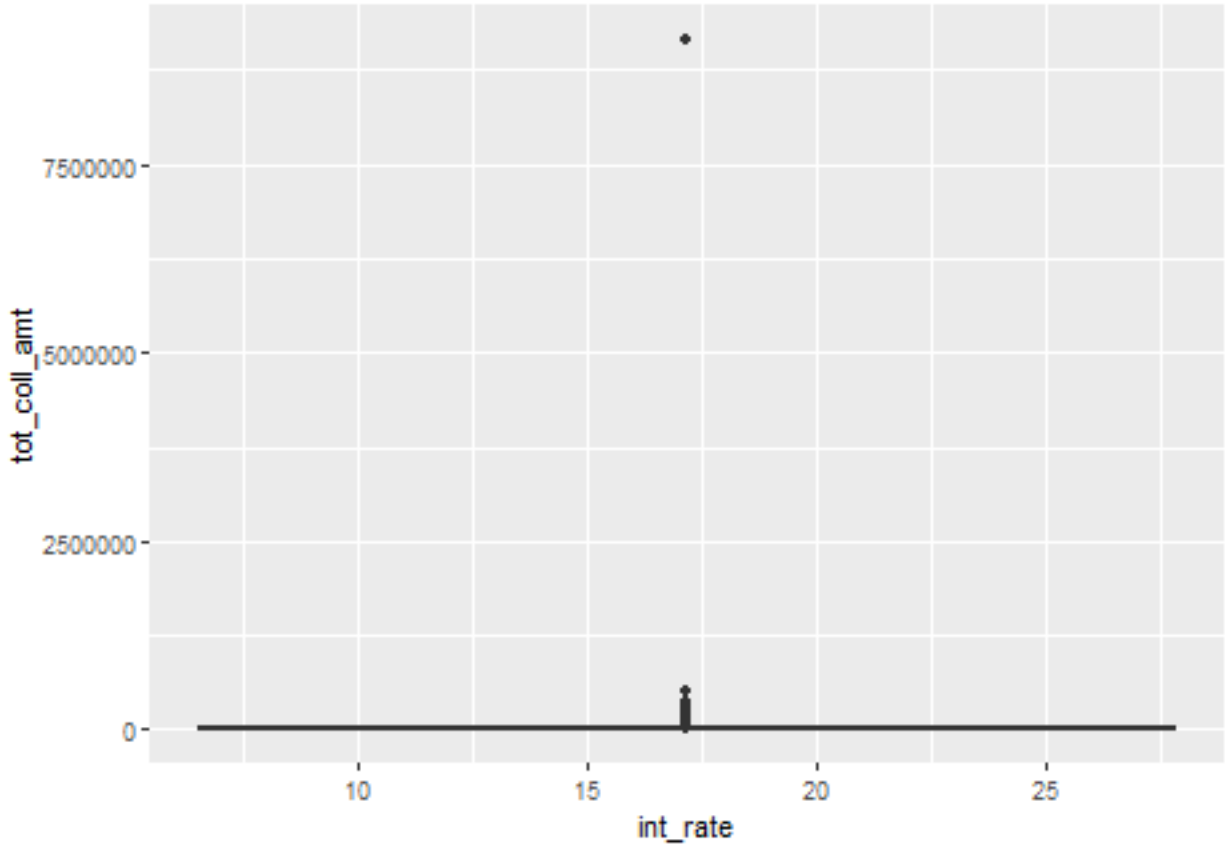
```
#Plotting uncleaned tot_coll_amt

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()`).
```

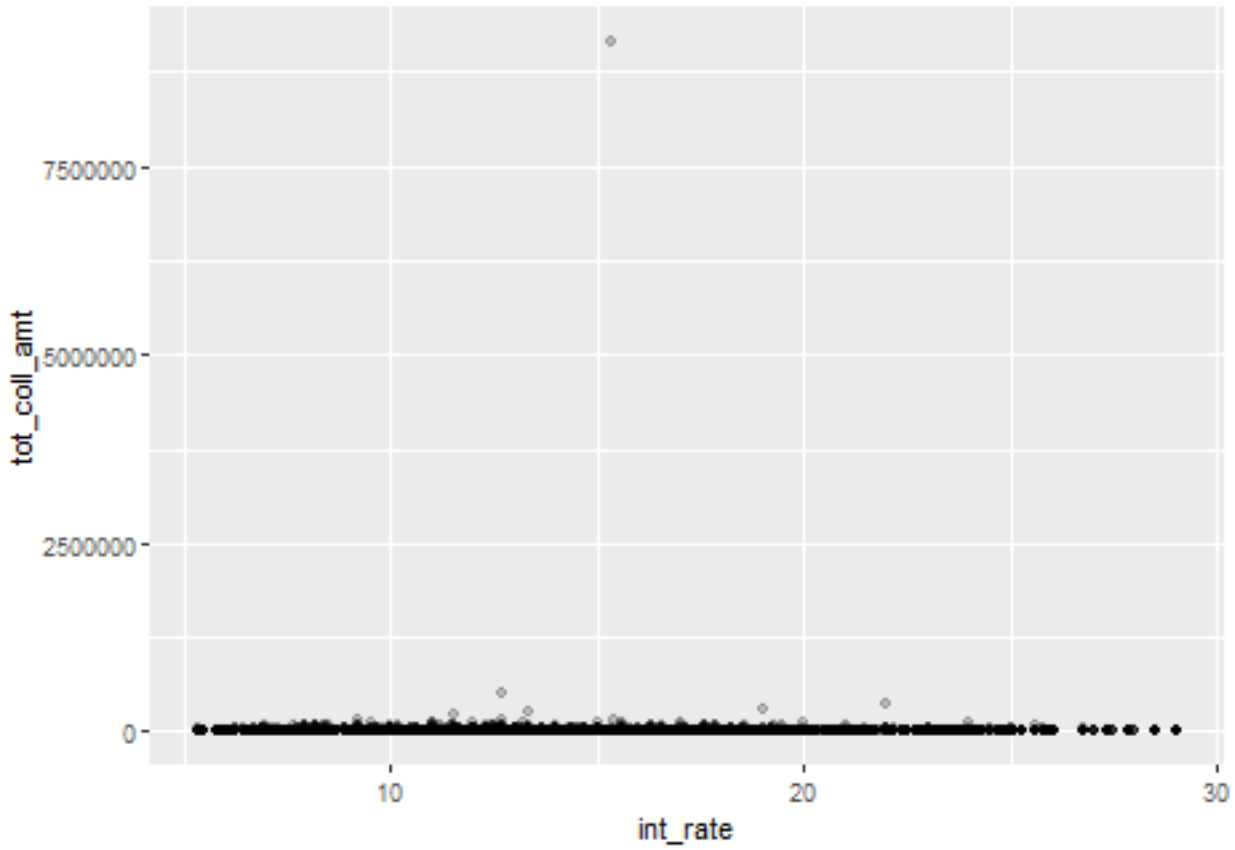



```
ggplot(data = cleaning, mapping = aes(x=int_rate,y=tot_coll_amt))+geom_boxplot()
```

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

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

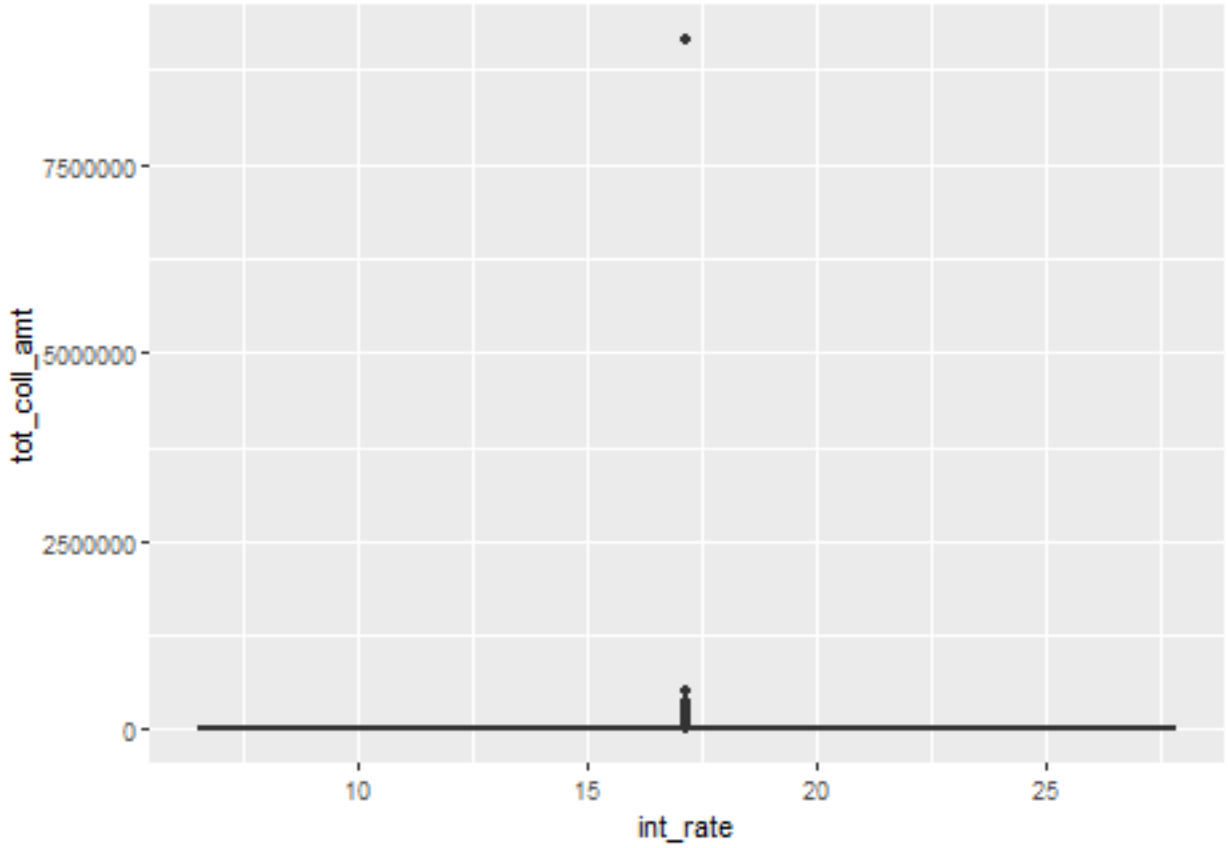



```
#Cleaning tot_coll_amt
cleaning <- cleaning %>% mutate(
    tot_coll_amt = ifelse(is.na(tot_coll_amt)== TRUE,0, tot_coll_amt))


#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()
```

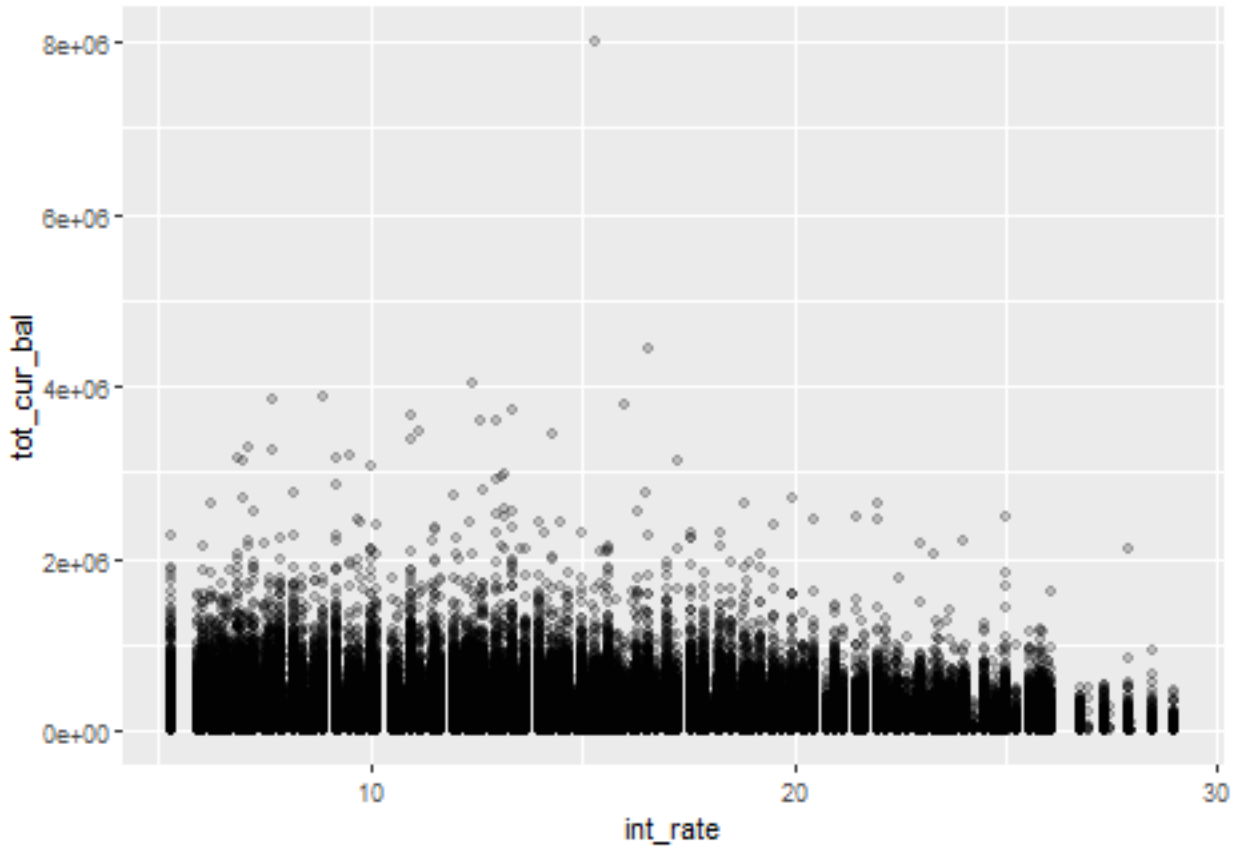
```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
```



#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)
```

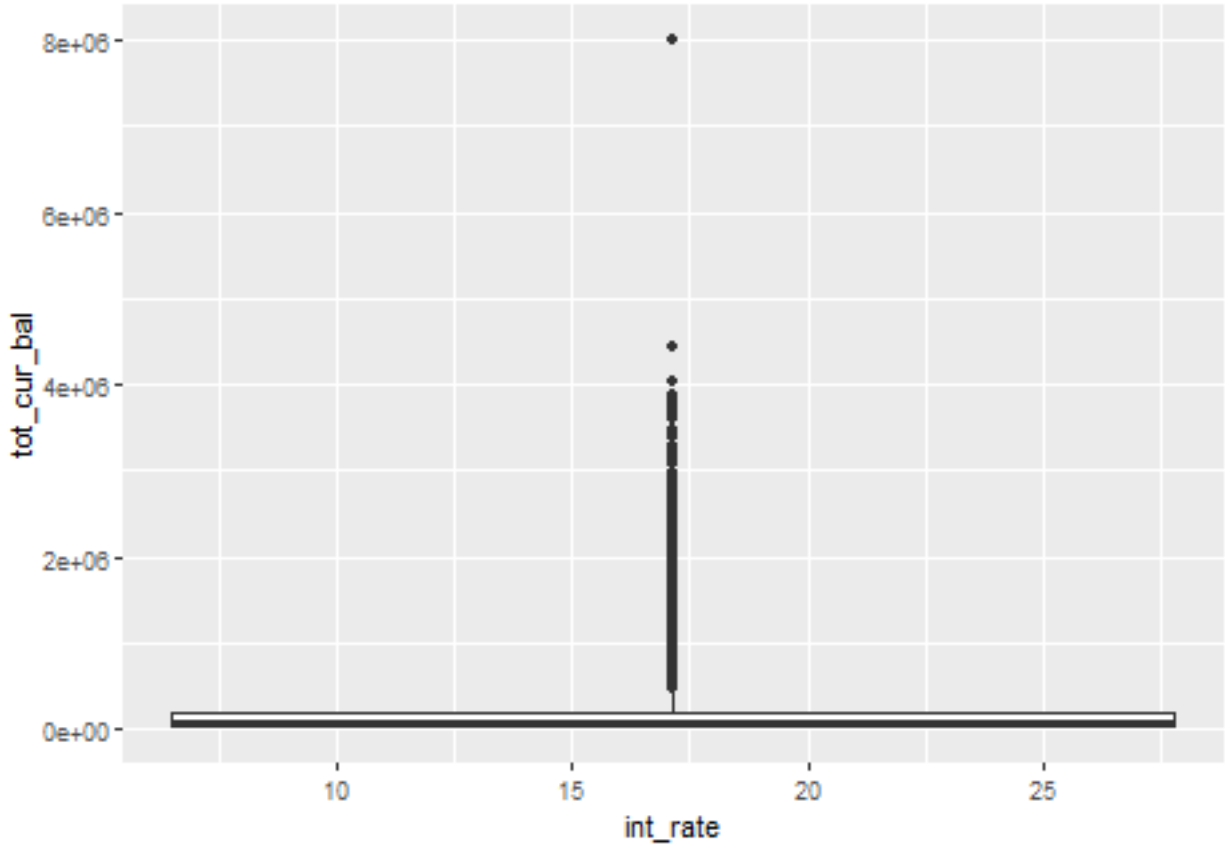
```
## Warning: Removed 63072 rows containing missing values (‘geom_point()‘).
```



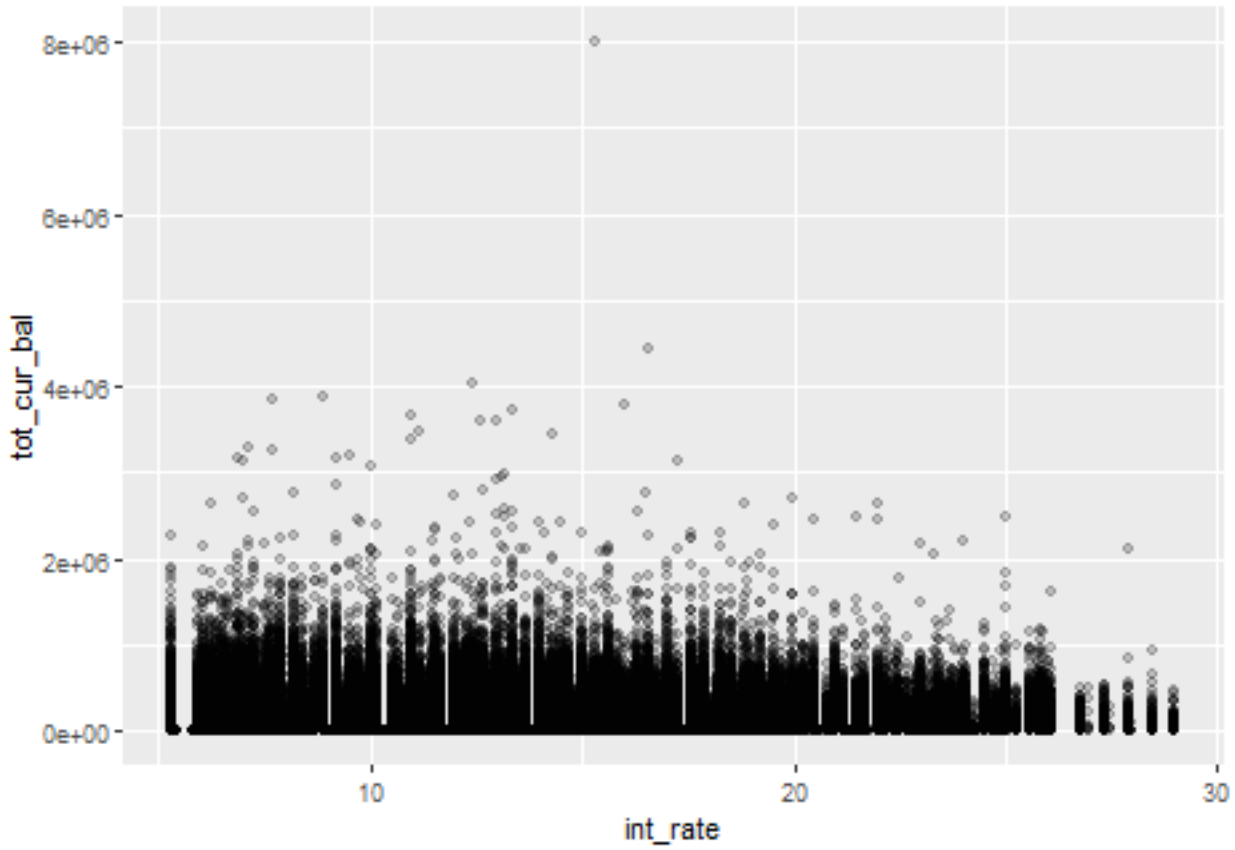

```
ggplot(data = cleaning, mapping = aes(x=int_rate,y=tot_cur_bal))+geom_boxplot()
```

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

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

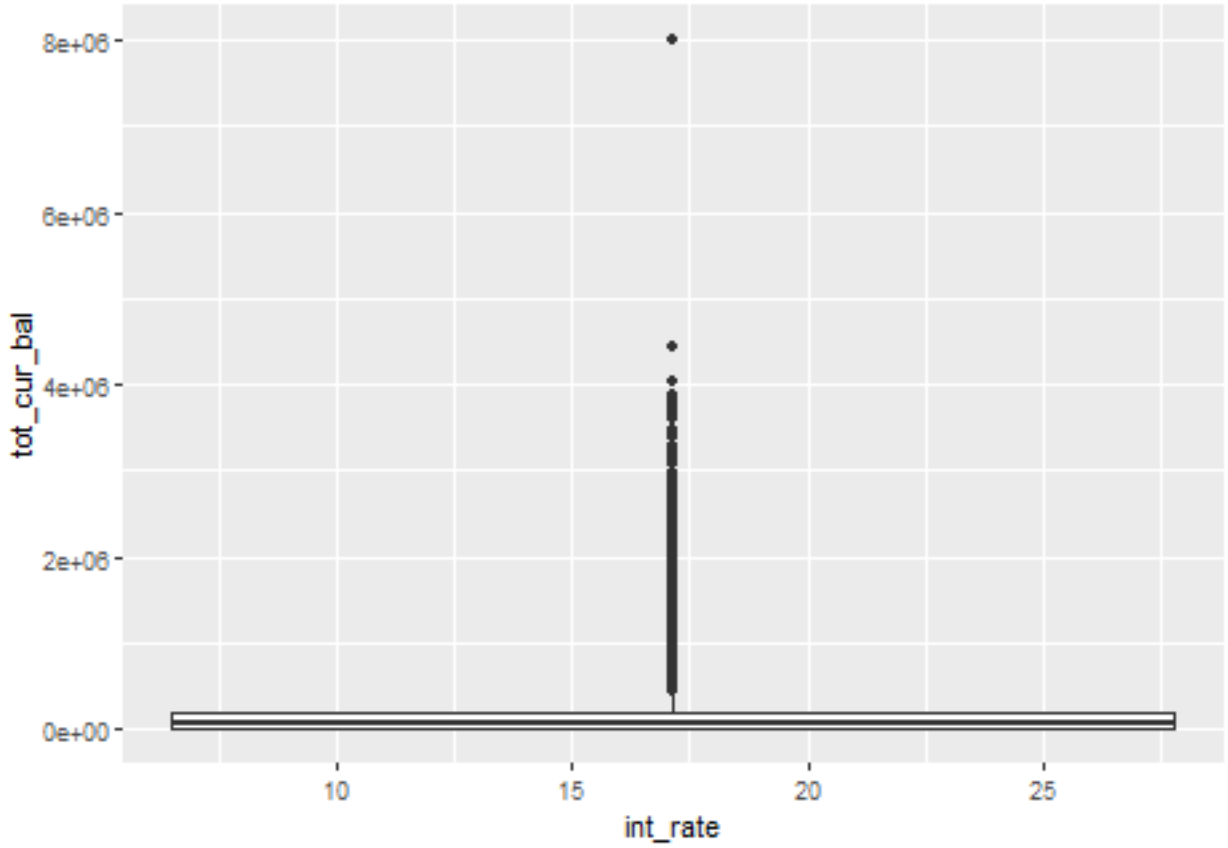



```
#Cleaning tot_cur_bal
cleaning <- cleaning %>% mutate(
    tot_cur_bal = ifelse(is.na(tot_cur_bal)== TRUE,0, tot_cur_bal))


#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 the ones with characters to factors

```
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)
```

By checking the summary again it's clear to see that there are only 460 (rows) joint applications. This number is too small compared to the total dataset of almost 800k rows. Also by joining the dti's the data that correlates to the interest is contained. Therefore we should just delete the columns verification_status_joint as well as application_type

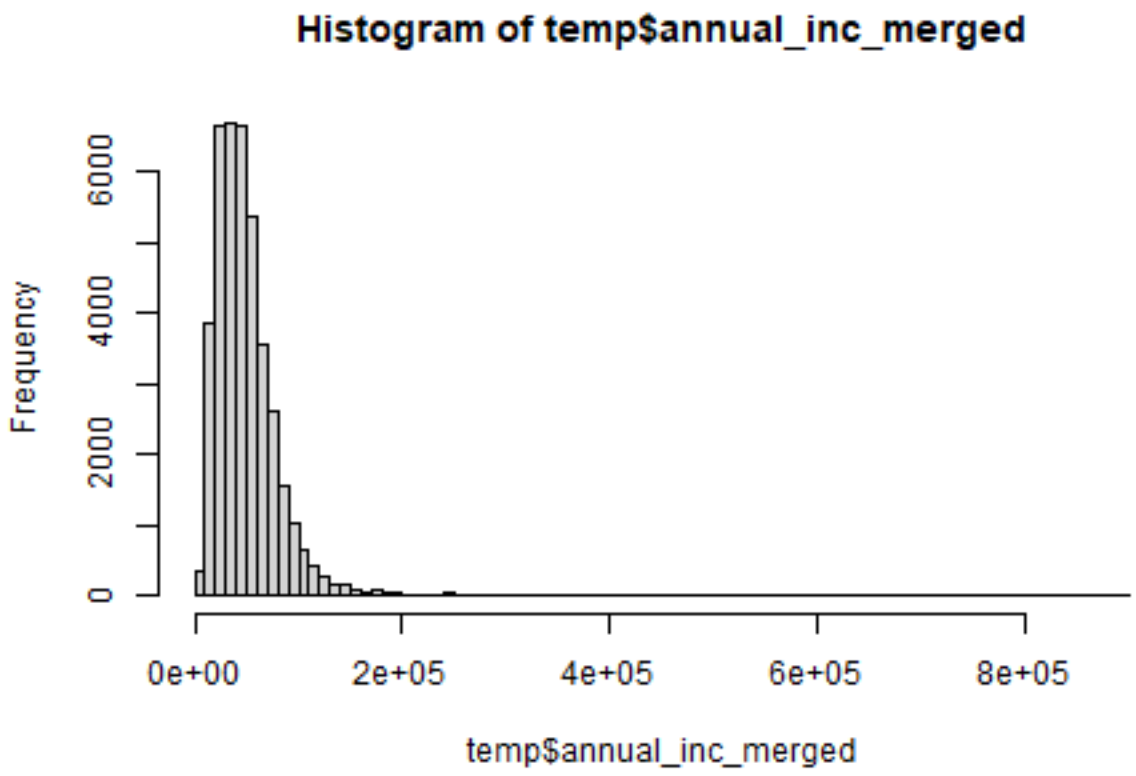
```
cleaning <- cleaning %>% select(-verification_status_joint, -application_type)
```

#Cleaning of emp_lenght

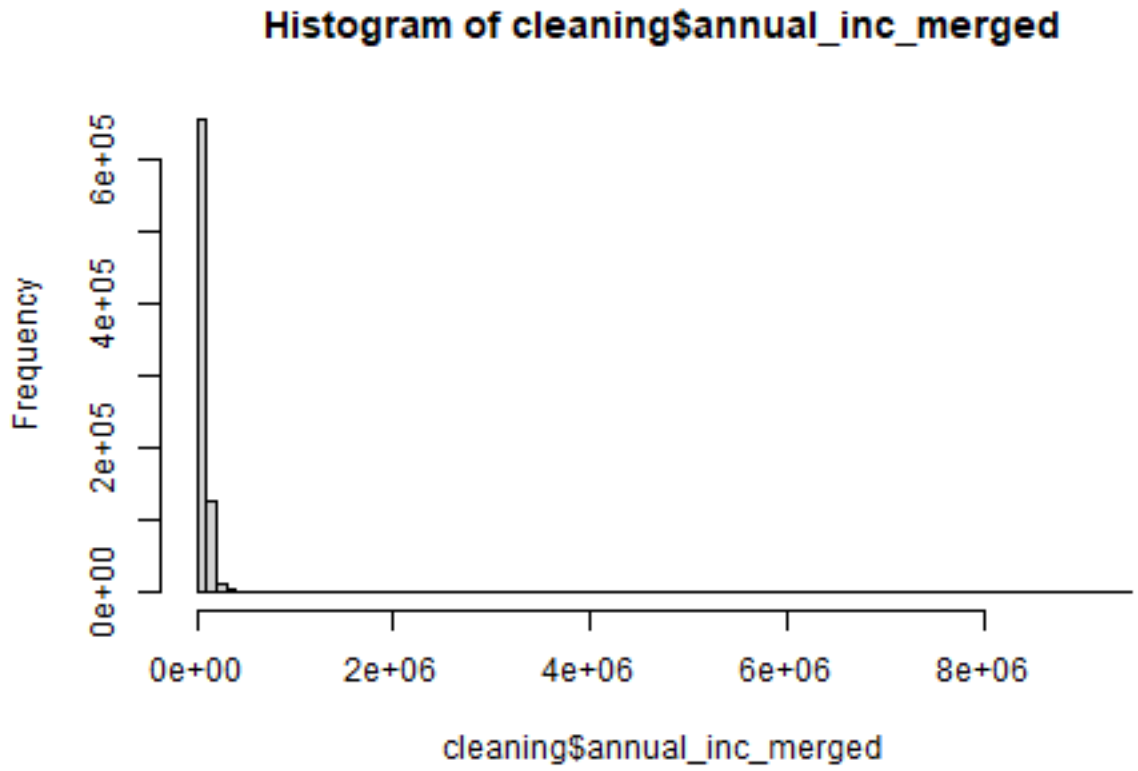
```
library(dplyr)

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
```

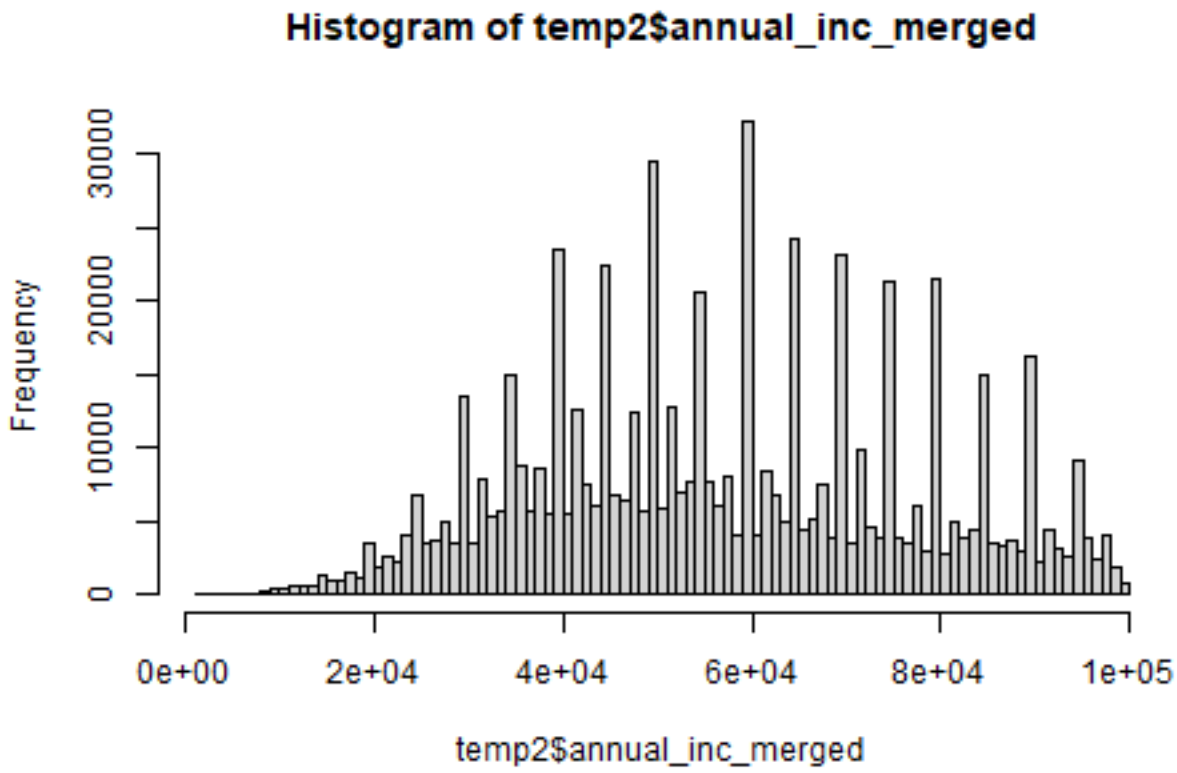
```
temp<-cleaning %>% filter(emp_length=="n/a")
hist(temp$annual_inc_merged,breaks = 100)
```



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



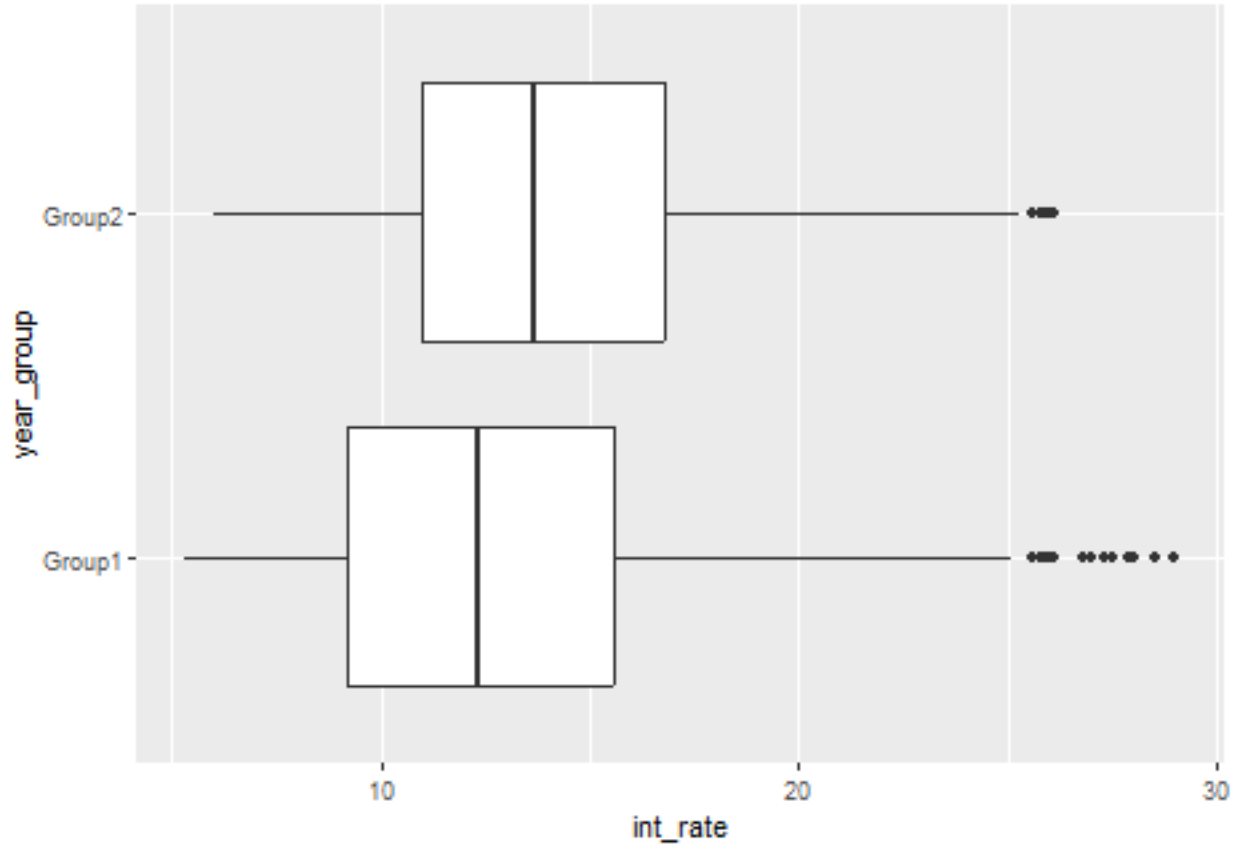

```
temp2<-cleaning %>% filter(annual_inc_merged<100000)
hist(temp2$annual_inc_merged,breaks = 100)
```




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

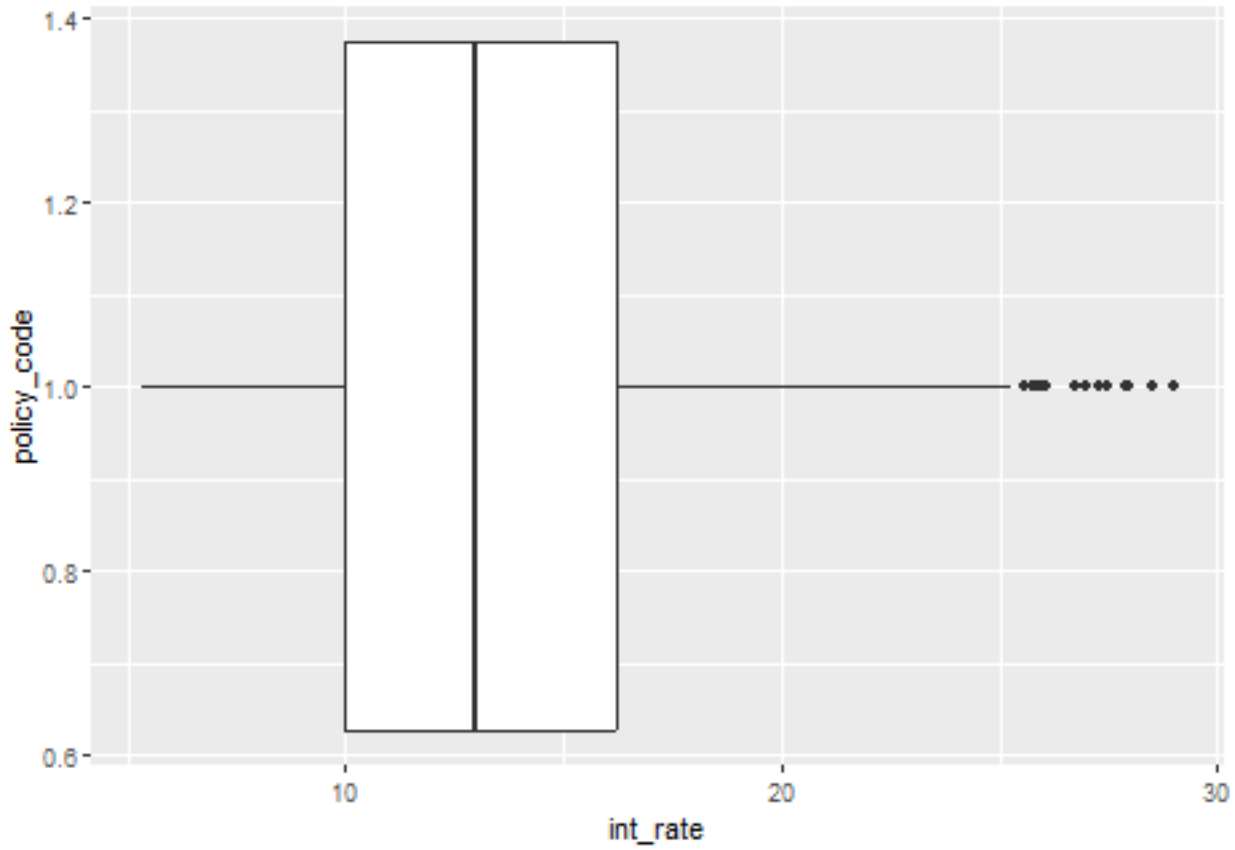
```
## [1] "2013" "2011" "2014" "2012" "2010" "2015" "2009" "2008" "2007"
```

```
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()
```



There is only policy code 1, therefore delete the column

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

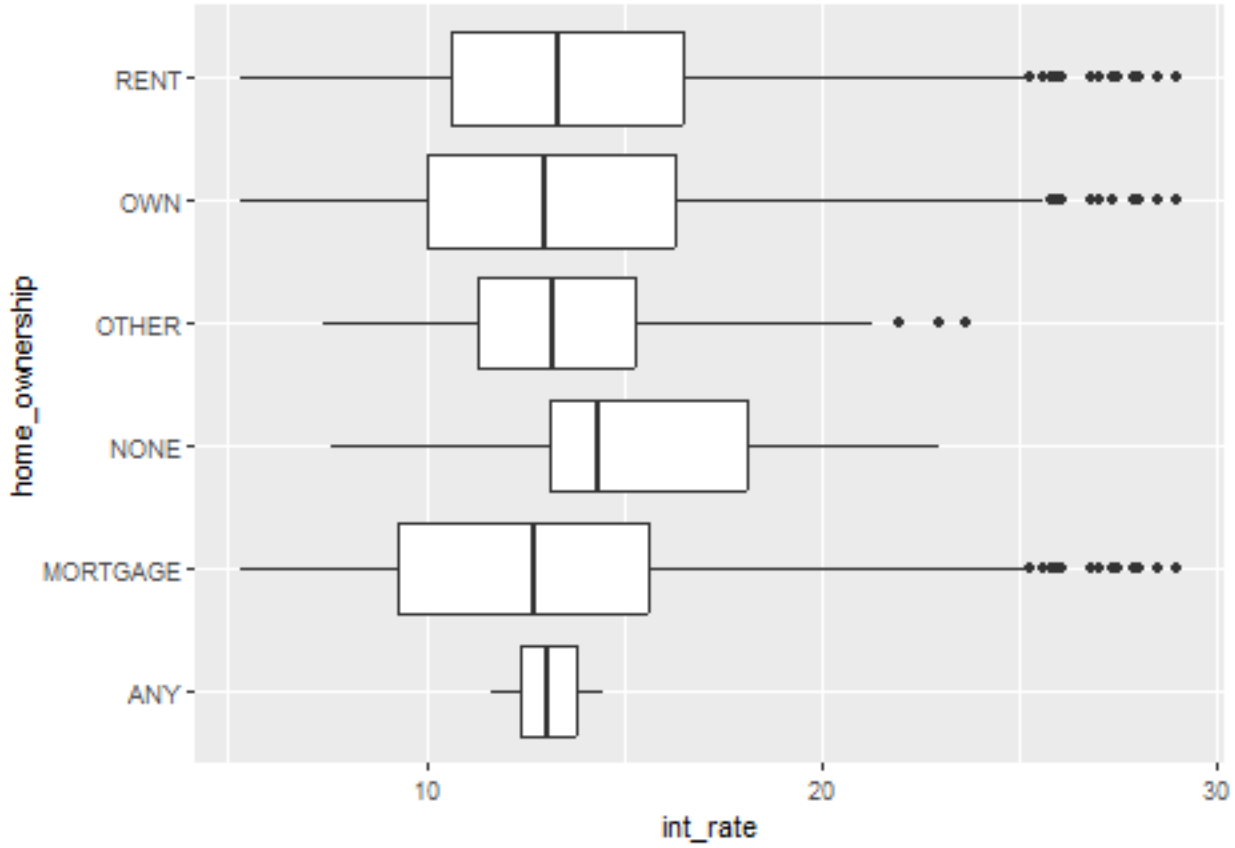



```
cleaning <- cleaning %>% select(-policy_code)
```

Cleaning of home_ownership When plotting the data, there seem to be no correlation to interest rates ANY are 2, OTHER are 154 and NONE are 39. Only none seem to have a higher interest rate then the others but with 39 cases this seems odd. Because None would mean they are homeless and we can not imagine giving loans to homeless people...

Factorize home ownership column after that code when rerunning because otherwise it will retain the deleted rows.

```
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

```
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)
```

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)
```

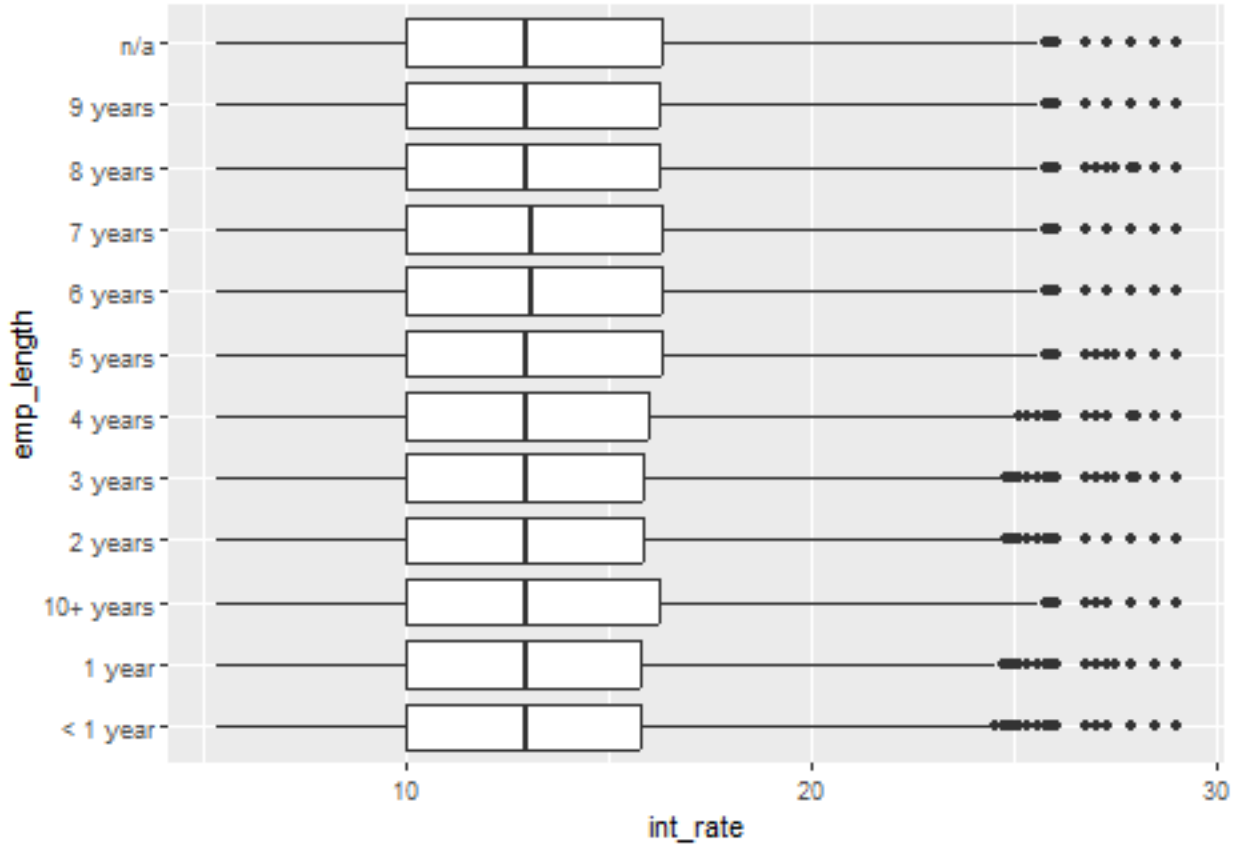
```
summary(cleaning)
```

```
##    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
##
##      emp_length      home_ownership        verification_status
##  10+ years:262163   MORTGAGE:398891   Not Verified   :240019
##  2 years  : 70881   OWN     : 78722   Source Verified:296478
##  < 1 year : 63362   RENT    :320278   Verified       :261394
##  3 years  : 62933
##  1 year   : 51468
##  5 years  : 50118
##  (Other)  :236966
##                purpose        delinq_2yrs      inq_last_6mths
##  debt_consolidation:471654   Min.   : 0.0000   Min.   : 0.0000
##  credit_card       :185353   1st Qu.: 0.0000   1st Qu.: 0.0000
##  home_improvement  : 46459   Median : 0.0000   Median : 0.0000
##  other             : 38611   Mean   : 0.3143   Mean   : 0.6945
##  major_purchase    : 15549   3rd Qu.: 0.0000   3rd Qu.: 1.0000
##  small_business    :  9349   Max.   :39.0000   Max.   :33.0000
##  (Other)           : 30916
##     open_acc        pub_rec          revol_bal         revol_util
##  Min.   : 1.00   Min.   : 0.0000   Min.   :      0   Min.   :  0.00
##  1st Qu.: 8.00   1st Qu.: 0.0000   1st Qu.:   6451   1st Qu.: 37.70
##  Median :11.00   Median : 0.0000   Median :  11882   Median : 56.00
##  Mean   :11.55   Mean   : 0.1954   Mean   :  16934   Mean   : 55.05
##  3rd Qu.:14.00   3rd Qu.: 0.0000   3rd Qu.:  20844   3rd Qu.: 73.50
##  Max.   :90.00   Max.   :63.0000   Max.   :2904836   Max.   :892.30
##
##    total_acc      initial_list_status   out_prncp     out_prncp_inv
##  Min.   :  1.00   f:410580            Min.   :    0   Min.   :    0
##  1st Qu.: 17.00   w:387311            1st Qu.:    0   1st Qu.:    0
##  Median : 24.00                       Median : 6465   Median : 6460
##  Mean   : 25.27                       Mean   : 8407   Mean   : 8403
##  3rd Qu.: 32.00                       3rd Qu.:13664   3rd Qu.:13660
##  Max.   :169.00                       Max.   :49373   Max.   :49373
```

```
##
##  collections_12_mths_ex_med acc_now_delinq       tot_coll_amt
##  Min.   : 0.00000           Min.   : 0.000000   Min.   :      0
##  1st Qu.: 0.00000           1st Qu.: 0.000000   1st Qu.:      0
##  Median : 0.00000           Median : 0.000000   Median :      0
##  Mean   : 0.01448           Mean   : 0.005026   Mean   :    210
##  3rd Qu.: 0.00000           3rd Qu.: 0.000000   3rd Qu.:      0
##  Max.   :20.00000           Max.   :14.000000   Max.   :9152545
##
##   tot_cur_bal       open_acc_6m         open_il_6m        open_il_12m
##  Min.   :      0   Min.   : 0.00000   Min.   : 0.00000   Min.   : 0.00000
##  1st Qu.:  23206   1st Qu.: 0.00000   1st Qu.: 0.00000   1st Qu.: 0.00000
##  Median :  65420   Median : 0.00000   Median : 0.00000   Median : 0.00000
##  Mean   : 128477   Mean   : 0.02641   Mean   : 0.06983   Mean   : 0.01817
##  3rd Qu.: 195890   3rd Qu.: 0.00000   3rd Qu.: 0.00000   3rd Qu.: 0.00000
##  Max.   :8000078   Max.   :14.00000   Max.   :33.00000   Max.   :12.00000
##
##   open_il_24m       mths_since_rcnt_il  total_bal_il         il_util
##  Min.   : 0.00000   Min.   :  0.0000   Min.   :     0.0   Min.   :  0.00
##  1st Qu.: 0.00000   1st Qu.:  0.0000   1st Qu.:     0.0   1st Qu.:  0.00
##  Median : 0.00000   Median :  0.0000   Median :     0.0   Median :  0.00
##  Mean   : 0.03992   Mean   :  0.4919   Mean   :   872.1   Mean   :  1.49
##  3rd Qu.: 0.00000   3rd Qu.:  0.0000   3rd Qu.:     0.0   3rd Qu.:  0.00
##  Max.   :19.00000   Max.   :363.0000   Max.   :878459.0   Max.   :223.30
##
##   open_rv_12m        open_rv_24m         max_bal_bc         all_util
##  Min.   : 0.00000   Min.   : 0.00000   Min.   :    0.0   Min.   :  0.000
##  1st Qu.: 0.00000   1st Qu.: 0.00000   1st Qu.:    0.0   1st Qu.:  0.000
##  Median : 0.00000   Median : 0.00000   Median :    0.0   Median :  0.000
##  Mean   : 0.03316   Mean   : 0.07115   Mean   :  140.8   Mean   :  1.457
##  3rd Qu.: 0.00000   3rd Qu.: 0.00000   3rd Qu.:    0.0   3rd Qu.:  0.000
##  Max.   :22.00000   Max.   :43.00000   Max.   :83047.0   Max.   :151.400
##
##  total_rev_hi_lim      inq_fi          total_cu_tl        inq_last_12m
##  Min.   :      0   Min.   : 0.00000   Min.   : 0.00000   Min.   :-4.00000
##  1st Qu.:  11700   1st Qu.: 0.00000   1st Qu.: 0.00000   1st Qu.: 0.00000
##  Median :  21800   Median : 0.00000   Median : 0.00000   Median : 0.00000
##  Mean   :  29568   Mean   : 0.02262   Mean   : 0.03669   Mean   : 0.04734
##  3rd Qu.:  37900   3rd Qu.: 0.00000   3rd Qu.: 0.00000   3rd Qu.: 0.00000
##  Max.   :9999999   Max.   :16.00000   Max.   :35.00000   Max.   :32.00000
##
##       mths_since_delinq_cat     mths_since_last_record_cat
##  1_to_3_years     :150675    1_to_3_years     : 11811
##  3_to_5_years     :100941    3_to_5_years     : 30524
##  more_than_5_years: 61595    more_than_5_years: 77818
##  No_delinq        :408518    No_record        :675618
##  recent           : 76162    recent           :  2120
##
##
##   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
##
```

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

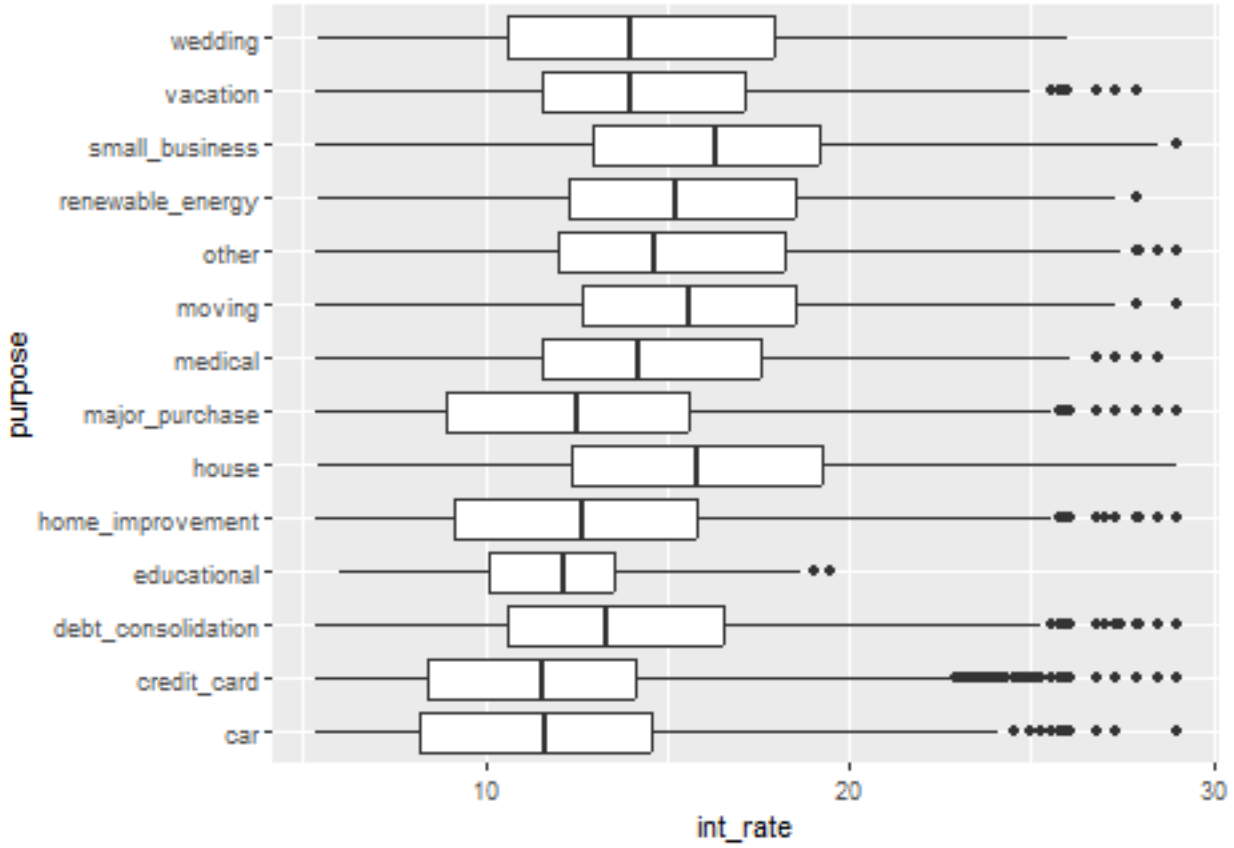


First we thought about cleaning emp_length because it still has 236966 n/a's. But after plotting emp_lenght it's clear that it does not have an impact on the interest. Therefore we delete this column.

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

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()
```



Make the final selection of attributes to consider for the model training and save the result to a file.

```
cleaning <- data.frame(cleaning %>% dplyr::select(int_rate,loan_amnt, term,installment,home_ownership, 
saveRDS(cleaning, "../../Data/Out/cleanData.rds")
```