numerical_feature_EDA_02

```
rm(list=ls())
loan <- read.csv('/Users/fanyang/Documents/lendingclub/2018_12_21/d2007_2015_loan.csv',
                 header = TRUE, stringsAsFactors = FALSE)
```

find numerical variables

```
numeric_fea <- colnames(loan)[which(sapply(loan, function(x) {return(is.numeric(x))}))]
numeric_fea
```

```
##  [1] "X"                           "loan_amnt"
##  [3] "funded_amnt"                 "funded_amnt_inv"
##  [5] "int_rate"                    "installment"
##  [7] "annual_inc"                  "dti"
##  [9] "delinq_2yrs"                 "inq_last_6mths"
## [11] "mths_since_last_delinq"      "mths_since_last_record"
## [13] "open_acc"                    "pub_rec"
## [15] "revol_bal"                   "revol_util"
## [17] "total_acc"                   "out_prncp"
## [19] "out_prncp_inv"               "total_pymnt"
## [21] "total_pymnt_inv"             "total_rec_prncp"
## [23] "total_rec_int"               "total_rec_late_fee"
## [25] "recoveries"                  "collection_recovery_fee"
## [27] "last_pymnt_amnt"             "collections_12_mths_ex_med"
## [29] "mths_since_last_major_derog" "policy_code"
## [31] "acc_now_delinq"              "tot_coll_amt"
## [33] "tot_cur_bal"                 "open_acc_6m"
## [35] "open_il_6m"                  "open_il_12m"
## [37] "open_il_24m"                 "mths_since_rcnt_il"
## [39] "total_bal_il"                "il_util"
## [41] "open_rv_12m"                 "open_rv_24m"
## [43] "max_bal_bc"                  "all_util"
## [45] "total_rev_hi_lim"            "inq_fi"
## [47] "total_cu_tl"                 "inq_last_12m"
## [49] "total_pymnt..79"             "last_pymnt_amnt..81"
## [51] "total_pymnt_inv..82"         "total_rec_int..83"
## [53] "total_rec_late_fee..84"      "total_rec_prncp..85"
## [55] "recoveries..86"              "collection_recovery_fee..87"
## [57] "out_prncp..88"               "out_prncp_inv..89"
## [59] "next_pymnt_binary"
```

count NA and drop columns that 80% data are NA

```
na_number <- sort((sapply(loan, function(x) {sum(is.na(x))})), decreasing = TRUE)
del_col <- names(na_number)[which(na_number > 0.8 * dim(loan)[1])]
del_col
```

```
##  [1] "il_util"                "mths_since_rcnt_il"
##  [3] "open_acc_6m"            "open_il_6m"
##  [5] "open_il_12m"            "open_il_24m"
##  [7] "total_bal_il"           "open_rv_12m"
##  [9] "open_rv_24m"            "max_bal_bc"
## [11] "all_util"               "inq_fi"
## [13] "total_cu_tl"            "inq_last_12m"
## [15] "mths_since_last_record"
```

```
dim(loan)
```

```
## [1] 601779     87
```

```
loan <- loan[, !(names(loan) %in% del_col)]
dim(loan)
```

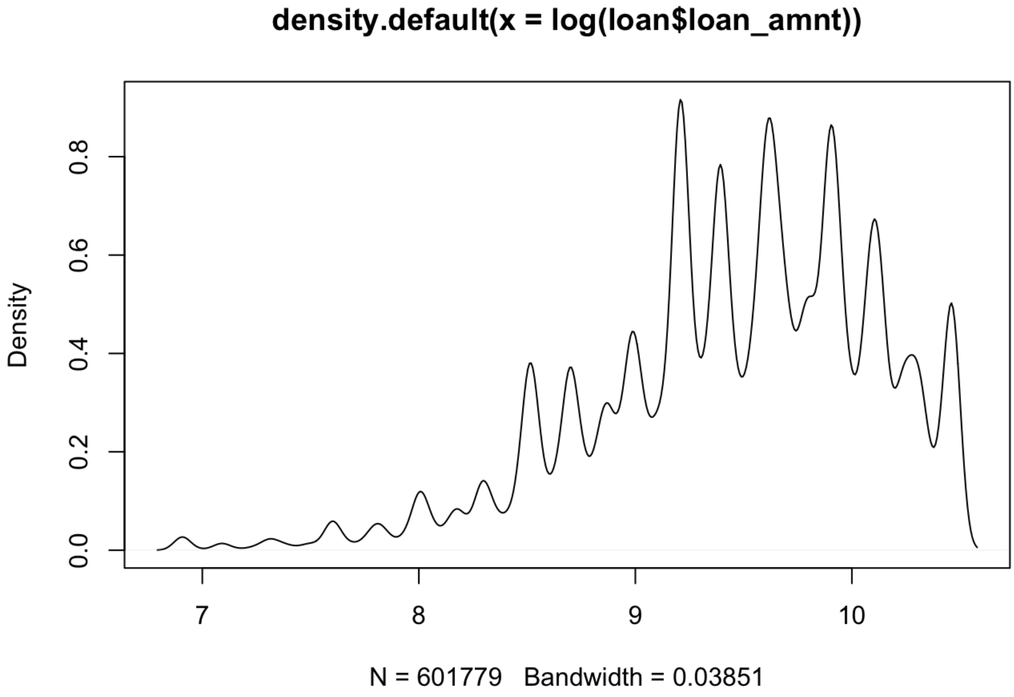
```
## [1] 601779     72
```

“loan_amnt”

```
summary(loan$loan_amnt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1000    8800   14000   15242   20000   35000
```

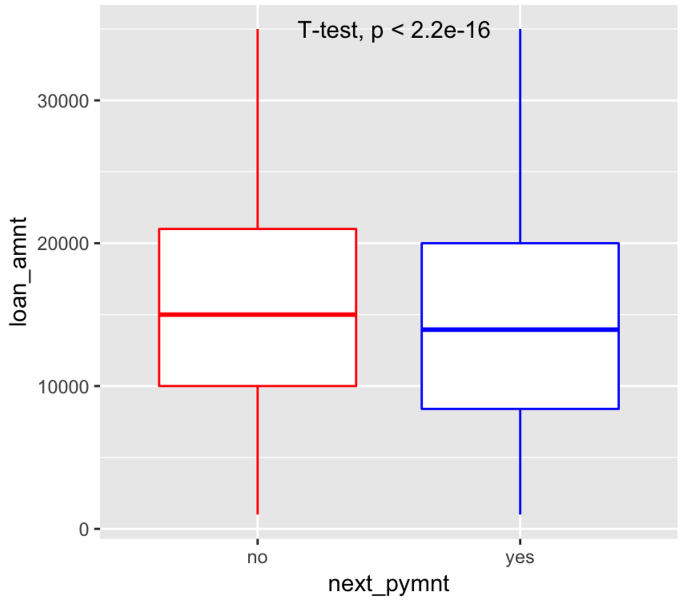
```
plot(density(log(loan$loan_amnt)))
```



```
library("ggplot2")
library("ggpubr")
```

```
## Loading required package: magrittr
```

```
loan$next_pymnt = ifelse(loan$next_pymnt_binary=='1', 'no', 'yes')
ggplot(data=loan, aes(x = next_pymnt, y = loan_amnt)) +
  geom_boxplot(color=c('red', 'blue')) +
  stat_compare_means(method = "t.test", label.x = 1.3, label.y = 35000)
```

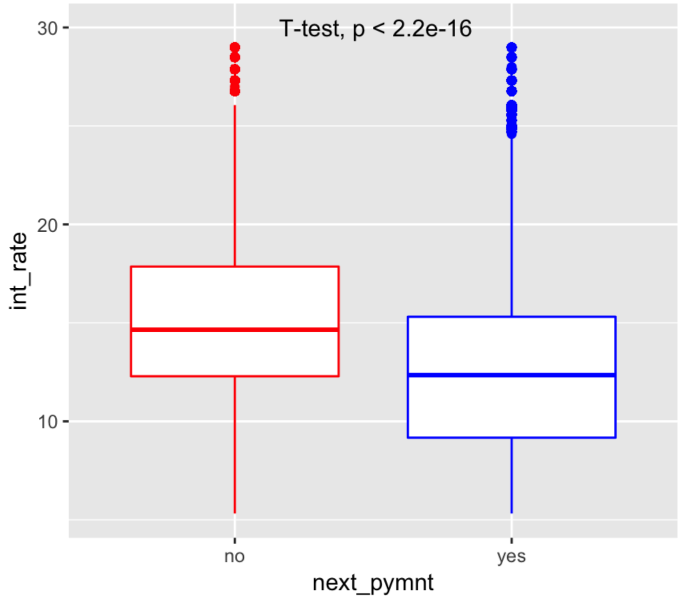


“int_rate”

```
summary(loan$int_rate)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    5.32    9.49   12.69   12.95   15.61   28.99
```

```
ggplot(loan, aes(x = next_pymnt, y = int_rate)) +
  geom_boxplot(color=c('red', 'blue')) +
  stat_compare_means(method = "t.test", label.x = 1.3, label.y = 30)
```

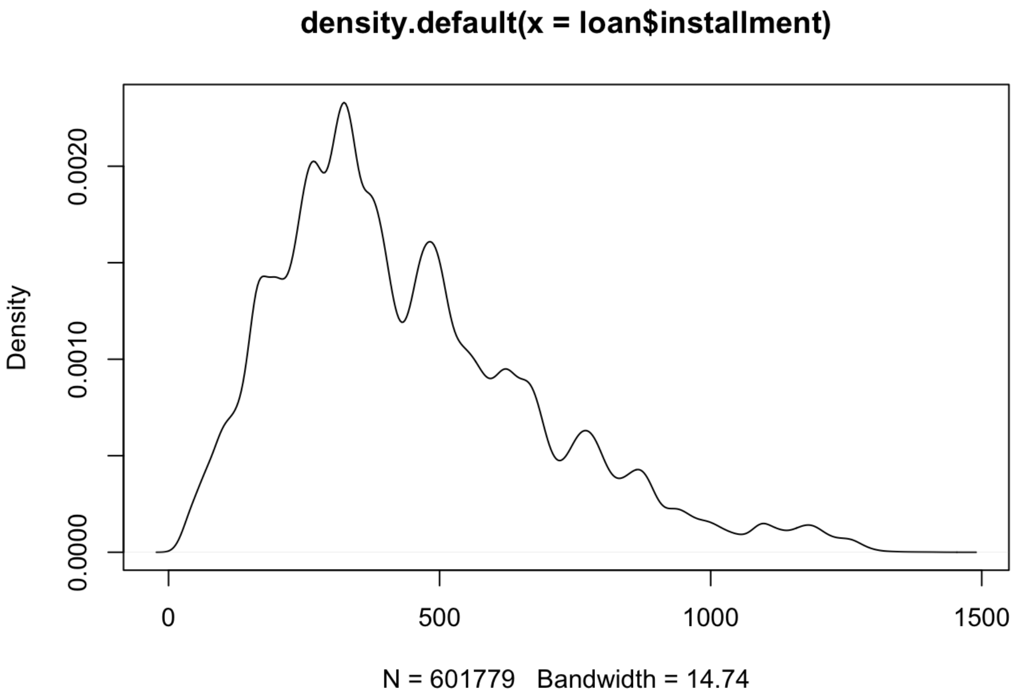


“installment”

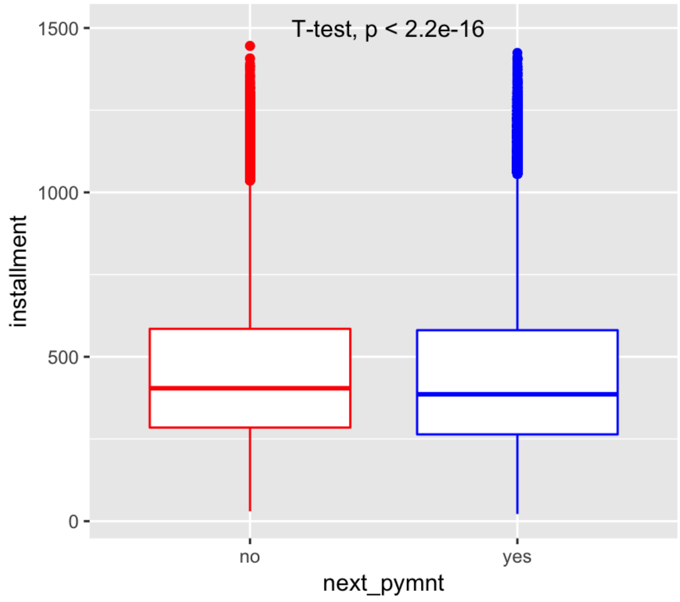
```
summary(loan$installment)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   21.74  267.21  389.01  443.83  581.45 1445.46
```

```
plot(density(loan$installment))
```



```
ggplot(loan, aes(x = next_pymnt, y = installment)) +
  geom_boxplot(color=c('red', 'blue')) +
  stat_compare_means(method = "t.test", label.x = 1.3, label.y = 1500)
```

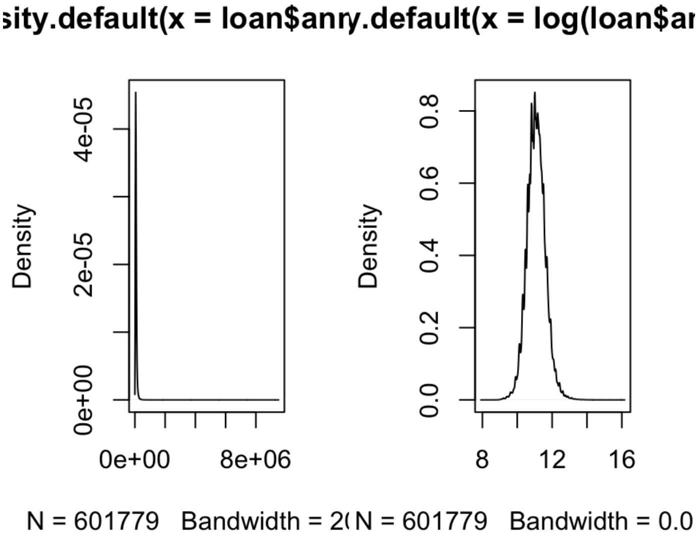


“annual_inc”

```
summary(loan$annual_inc)
```

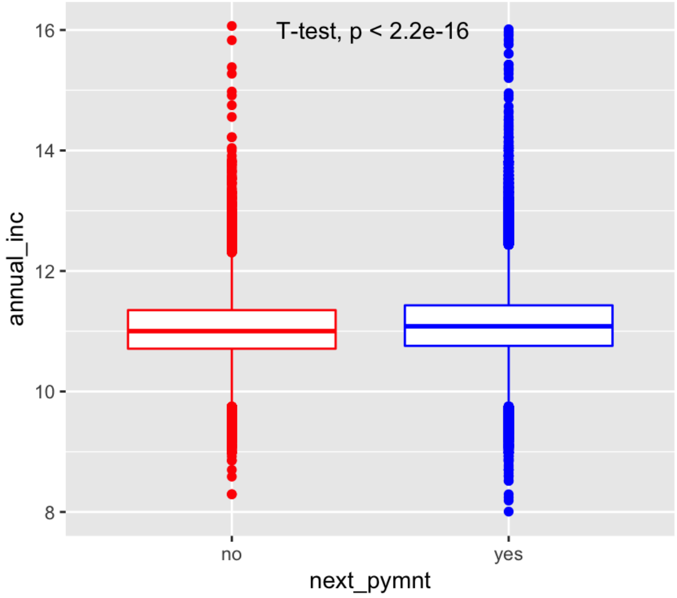
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3000   46000   65000   76189   90000 9500000
```

```
par(mfrow=c(1,2))
plot(density(loan$annual_inc))
plot(density(log(loan$annual_inc)))
```



```
# transform 'annual_income' by taking log
loan$annual_inc <- log(loan$annual_inc+1)
```

```
ggplot(loan, aes(x=next_pymnt, y=annual_inc)) +
  geom_boxplot(color=c('red', 'blue')) +
  stat_compare_means(method = "t.test", label.x = 1.3, label.y = 16)
```



```
# 'annual_inc' has outlier, I will divide it into bins
loan$income_level = ifelse(loan$annual_inc <= quantile(loan$annual_inc, 0.01), 'exlow',
                           ifelse(loan$annual_inc <= quantile(loan$annual_inc, 0.25), 'low',
                                  ifelse(loan$annual_inc <= quantile(loan$annual_inc, 0.5), 'lowmedium',
                                         ifelse(loan$annual_inc <= quantile(loan$annual_inc, 0.75), 'medi
umhigh',
                                                ifelse(loan$annual_inc <= quantile(loan$annual_inc, 0.99)
, 'high',
                                                              'exhigh')))))
```

```
sort(table(loan$income_level))
```

```
## 
##     exhigh      exlow mediumhigh       high        low  lowmedium 
##       5853       6097     140100     143693     145926     160110
```

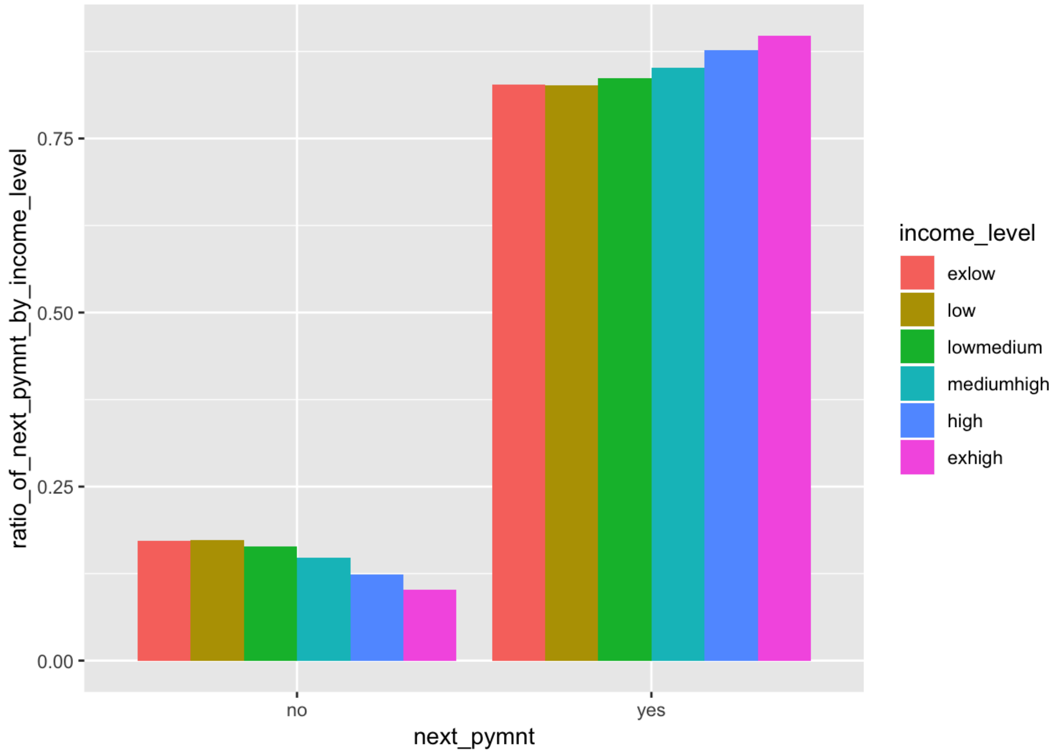
```
with(loan, table(income_level, next_pymnt)) / as.numeric(table(loan$income_level))
```

```
##             next_pymnt
## income_level        no       yes
##   exhigh     0.1019990 0.8980010
##   exlow      0.1725439 0.8274561
##   high       0.1232141 0.8767859
##   low        0.1736360 0.8263640
##   lowmedium  0.1637499 0.8362501
##   mediumhigh 0.1480514 0.8519486
```

```
d <- data.frame(with(loan, table(income_level, next_pymnt)) / as.numeric(table(loan$income_level)))
colnames(d)[3]<- c('ratio_of_next_pymnt_by_income_level')

d$income_level <- factor(d$income_level, levels = c('exlow', 'low', 'lowmedium', 'mediumhigh', 'high', 'exhigh'))

ggplot(data = d, aes(x=next_pymnt, y=ratio_of_next_pymnt_by_income_level, fill=income_level))+
  geom_bar(stat = "identity", position = position_dodge())
```

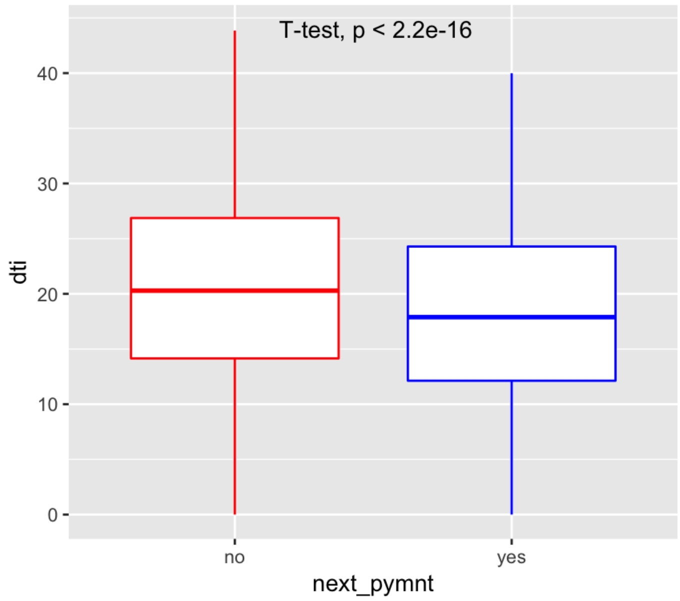


“dti”

```
summary(loan$dti)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   12.40   18.24   18.74   24.72   43.86
```

```
ggplot(loan, aes(x=next_pymnt, y=dti)) +
  geom_boxplot(color=c('red', 'blue')) +
  stat_compare_means(method = "t.test", label.x = 1.3, label.y = 44)
```

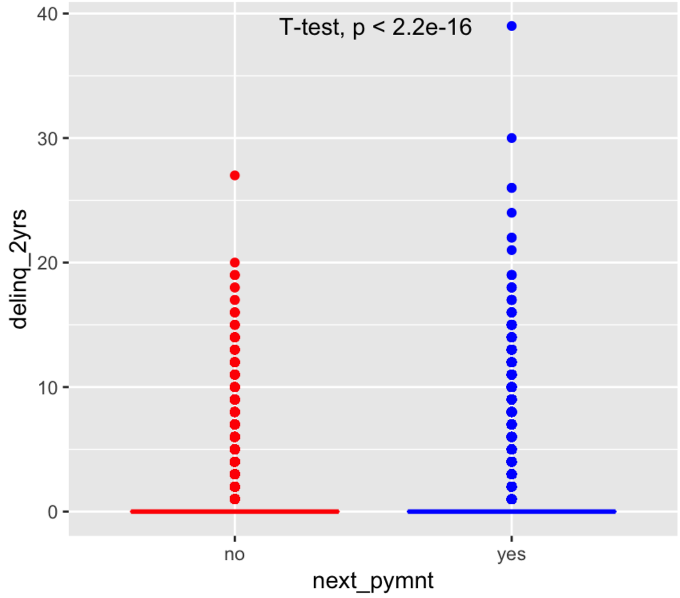


“delinq_2yrs”

```
summary(loan$delinq_2yrs)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.0000  0.0000  0.3374  0.0000 39.0000
```

```
ggplot(loan, aes(x=next_pymnt, y=delinq_2yrs)) +
  geom_boxplot(color=c('red', 'blue')) +
  stat_compare_means(method = "t.test", label.x = 1.3, label.y = 39)
```



```
# this feature is very skewed suggesting most people does not have delinq.
# I will generate a binary feature for it
loan$delinq_binary = ifelse(loan$delinq_2yrs==0, 'no', 'yes')

sort(table(loan$delinq_binary))
```

```
## 
##    yes     no 
## 122304 479475
```

```
with(loan, table(delinq_binary, next_pymnt)) / as.numeric(table(loan$delinq_binary))
```

```
##              next_pymnt
## delinq_binary        no       yes
##           no  0.1503603 0.8496397
##           yes 0.1599130 0.8400870
```

“inq_last_6mths”

```
summary(loan$inq_last_6mths)
```

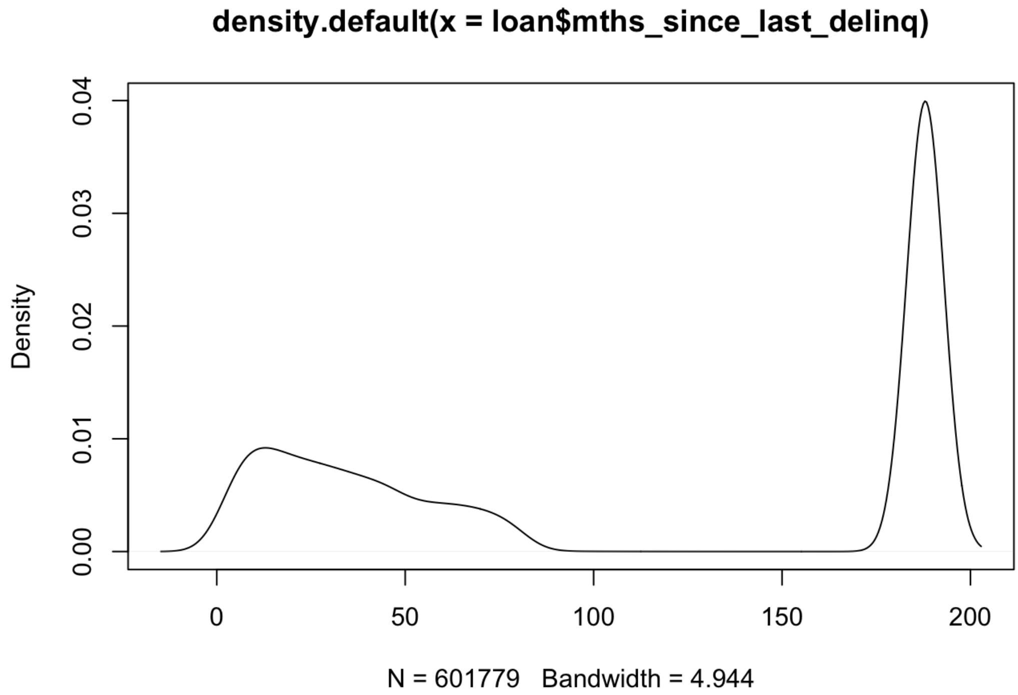
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.0000  0.0000  0.6084  1.0000  8.0000
```

“mths_since_last_delinq”

```
summary(loan$mths_since_last_delinq)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.00   15.00   30.00   33.77   49.00  188.00  298366
```

```
# NA probably means there is no delinq
# impute missing value with its maxium value
loan$mths_since_last_delinq[which(is.na(loan$mths_since_last_delinq))] = 188
plot(density(loan$mths_since_last_delinq))
```



“open_acc”

```
summary(loan$open_acc)
```

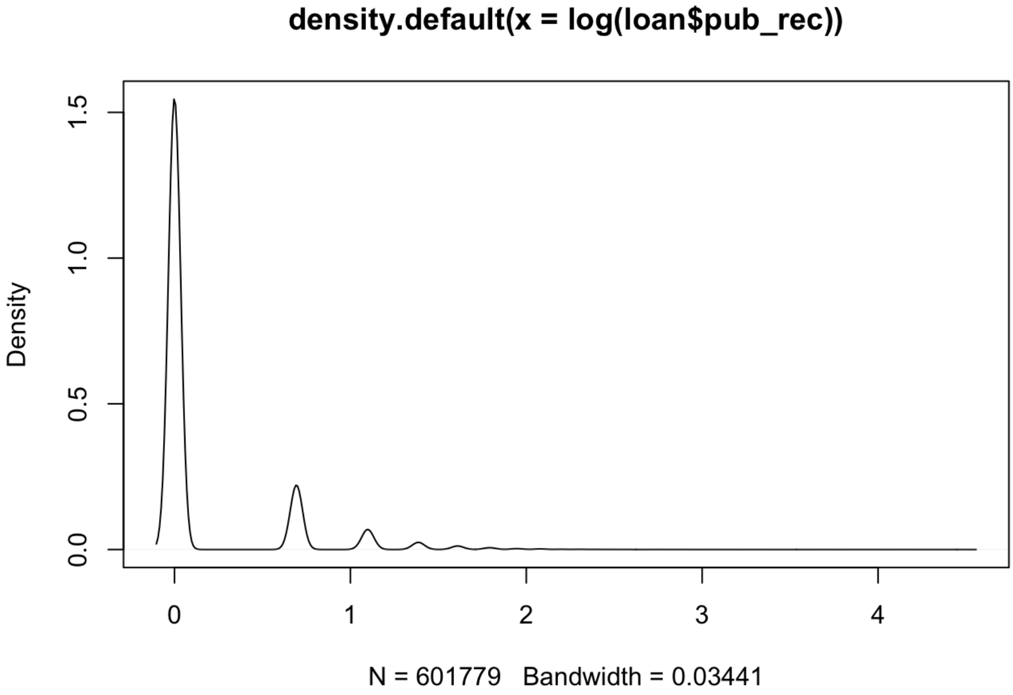
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     0.0     8.0    11.0    11.8    15.0    90.0
```

“pub_rec”

```
summary(loan$pub_rec)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   0.000   0.000   0.216   0.000  86.000
```

```
plot(density(log(loan$pub_rec)))
```



“revol_bal”: Total credit revolving balance

```
summary(loan$revol_bal)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0    6742   12337   17646   21647 2904836
```

“total_acc”

```
summary(loan$total_acc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    2.00   17.00   24.00   25.37   32.00  169.00
```

“total_pymnt” —> pymnt_percentaget = pymnt/loan_amount

```
loan$pymnt_pct = loan$total_pymnt/loan$loan_amnt
summary(loan$pymnt_pct)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.1245  0.2662  0.3490  0.5110  1.5902
```

“total_rec_late_fee”

```
summary(loan$total_rec_late_fee)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##   0.0000   0.0000   0.0000   0.1502   0.0000 252.8000
```

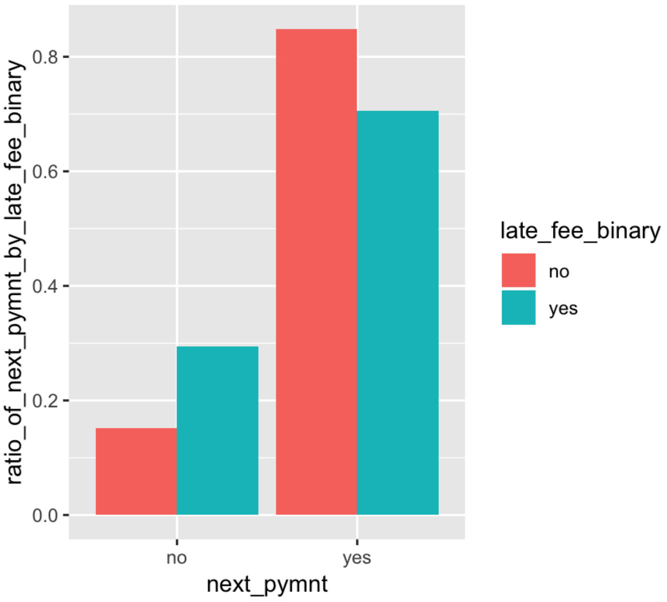
```
# generate a binary feature detechting whether customer had late fee for the loan
loan$late_fee_binary = ifelse(loan$total_rec_late_fee == 0, 'no', 'yes')
sort(table(loan$late_fee_binary))
```

```
##
##    yes     no
##   3352 598427
```

```
with(loan, table(late_fee_binary, next_pymnt)) / as.numeric(table(loan$late_fee_binary))
```

```
##                  next_pymnt
## late_fee_binary         no       yes
##              no  0.1515039 0.8484961
##              yes 0.2947494 0.7052506
```

```
d <- data.frame(with(loan, table(late_fee_binary, next_pymnt)) / as.numeric(table(loan$late_fee_binary)))
colnames(d)[3]<- c('ratio_of_next_pymnt_by_late_fee_binary')
ggplot(data = d, aes(x=next_pymnt, y=ratio_of_next_pymnt_by_late_fee_binary, fill=late_fee_binary))+
  geom_bar(stat = "identity", position = position_dodge())
```



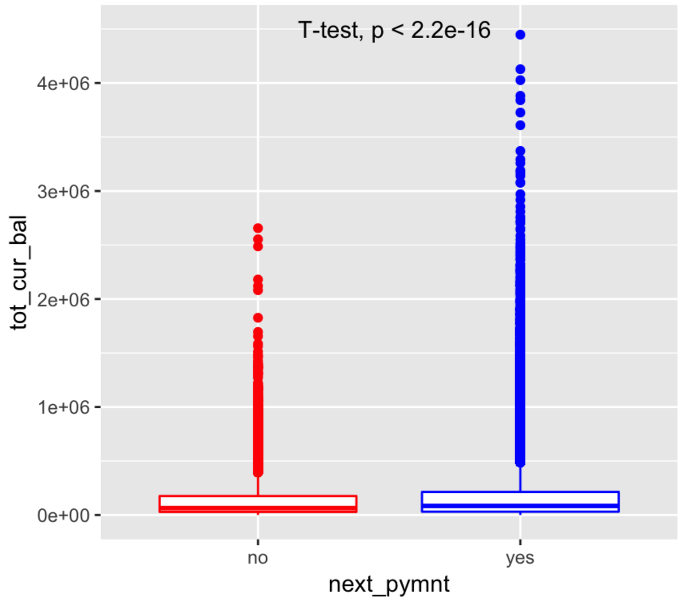
“tot_cur_bal”

```
summary(loan$tot_cur_bal)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0   30367   81079  140278  208952 4447397    3587
```

```
# impute NA with median
loan$tot_cur_bal[which(is.na(loan$tot_cur_bal))] = median(loan$tot_cur_bal, na.rm = TRUE)
```

```
ggplot(loan, aes(x=next_pymnt, y=tot_cur_bal)) +
  geom_boxplot(color=c('red', 'blue')) +
  stat_compare_means(method = "t.test", label.x = 1.3, label.y = 4500000 )
```



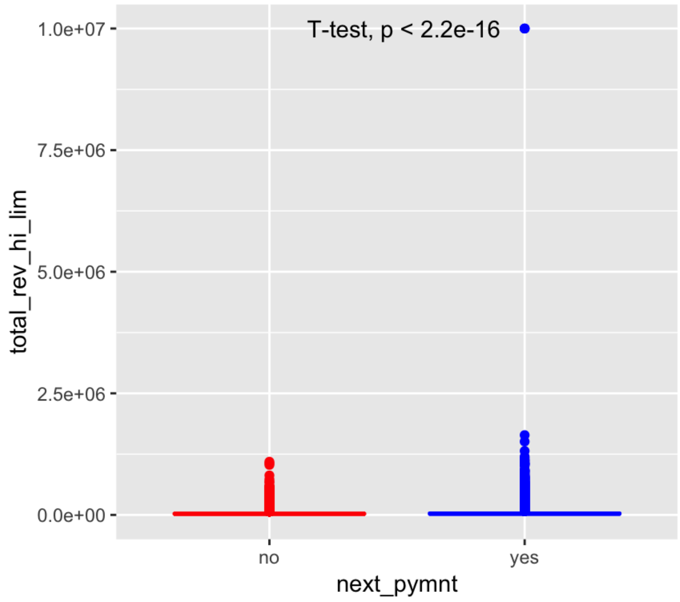
“total_rev_hi_lim”

```
summary(loan$total_rev_hi_lim)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##       0   14200   24300   32950   40900 9999999    3587
```

```
# impute NA with median
loan$total_rev_hi_lim[which(is.na(loan$total_rev_hi_lim))] = median(loan$total_rev_hi_lim, na.rm = TRUE)
```

```
ggplot(loan, aes(x=next_pymnt, y=total_rev_hi_lim)) +
  geom_boxplot(color=c('red', 'blue')) +
  stat_compare_means(method = "t.test", label.x = 1.3, label.y = 9999999)
```



numerical features and derivatives will be selected for prediction model, including:

```
# notice some numerical features have outliers
# 'loan_amnt', 'int_rate', 'installment', 'annual_inc', 'dti', 'delinq_2yrs', 'inq_last_6mths', 'mths_since_last_delinq',
# 'open_acc', 'pub_rec', 'revol_bal', 'total_acc', 'pymnt_pct', 'tot_cur_bal', 'total_rev_hi_lim'
# 'income_level', 'delinq_binary', 'late_fee_binary'
```

unuseful numerical features to remove, including:

```
# 'open_acc', 'revol_util', 'out_prncp', 'total_pymnt_inv', 'total_rec_prncp', 'mths_since_last_major_derog',
# 'total_rec_int', 'recoveries', 'collection_recovery_fee', 'policy_code',
# 'open_il_6m', 'open_il_24m', 'open_acc_6m', 'collections_12_mths_ex_med', 'acc_now_delinq',
# 'open_il_12m', 'mths_since_last_major_derog', 'tot_coll_amt', 'last_pymnt_amnt', 'mths_since_rcnt_il', 'total_bal_il',
# 'total_bal_il', 'open_rv_12m', 'open_rv_24m', 'max_bal_bc', 'all_util', 'inq_fi', 'total_cu_tl', 'inq_last_12m'
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.