Fan_Yang Q3

Fan Yang

April 28, 2018

import data

```
loan <- read.csv("loan.csv", stringsAsFactors = FALSE)
loanT <- loan
```

discard features that have over 80% missing value

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

split train and test data by 0.7 ratio

```
set.seed(1)
train.ind <- sample(1:dim(loan)[1], 0.7 * dim(loan)[1])
train <- loan[train.ind, ]
test <- loan[-train.ind, ]
```

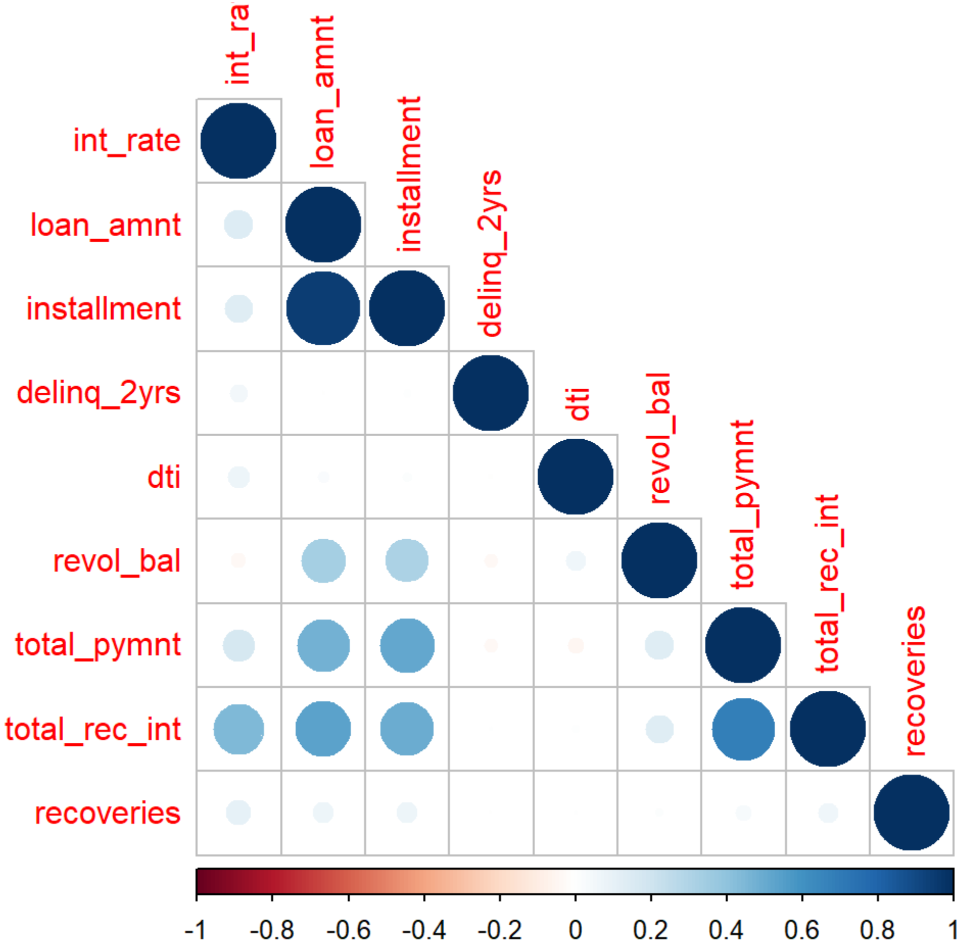
```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
correlations <- cor(loan[, c('int_rate', 'loan_amnt', 'installment', 'delinq_2yrs',
'dti', 'revol_bal', 'total_pymnt', 'total_rec_int', 'recoveries')], use = "pairwise
.complete.obs")
correlations
```

```
##                  int_rate     loan_amnt installment   delinq_2yrs
## int_rate       1.00000000  0.1450230993 0.133074919  0.0551777715
## loan_amnt      0.14502310  1.0000000000 0.944977241 -0.0001586672
## installment    0.13307492  0.9449772408 1.000000000  0.0078759022
## delinq_2yrs    0.05517777 -0.0001586672 0.007875902  1.0000000000
## dti            0.07990255  0.0206754869 0.014332840 -0.0028305679
## revol_bal     -0.03570809  0.3335799918 0.316588187 -0.0324862597
## total_pymnt    0.17050629  0.4746259450 0.514953671 -0.0319022162
## total_rec_int  0.44567882  0.5339157313 0.498664643  0.0016651105
## recoveries     0.10683996  0.0731500881 0.074546343 -0.0007291773
##                        dti   revol_bal total_pymnt total_rec_int
## int_rate       0.079902551 -0.03570809  0.17050629   0.445678819
## loan_amnt      0.020675487  0.33357999  0.47462594   0.533915731
## installment    0.014332840  0.31658819  0.51495367   0.498664643
## delinq_2yrs   -0.002830568 -0.03248626 -0.03190222   0.001665111
## dti            1.000000000  0.06727728 -0.04152877   0.008379887
## revol_bal      0.067277283  1.00000000  0.13832761   0.137379653
## total_pymnt   -0.041528769  0.13832761  1.00000000   0.681665949
## total_rec_int  0.008379887  0.13737965  0.68166595   1.000000000
## recoveries     0.001161910  0.01082837  0.03836135   0.067777247
##                   recoveries
## int_rate       0.1068399590
## loan_amnt      0.0731500881
## installment    0.0745463430
## delinq_2yrs   -0.0007291773
## dti            0.0011619099
## revol_bal      0.0108283664
## total_pymnt    0.0383613462
## total_rec_int  0.0677772470
## recoveries     1.0000000000
```

```
corrplot(correlations, method = "circle", tl.cex = 1, type = 'lower')
```



visulization of int_rate

```
mean(loan$int_rate)
```

```
## [1] 13.24674
```

```
sd(loan$int_rate)
```

```
## [1] 4.381867
```

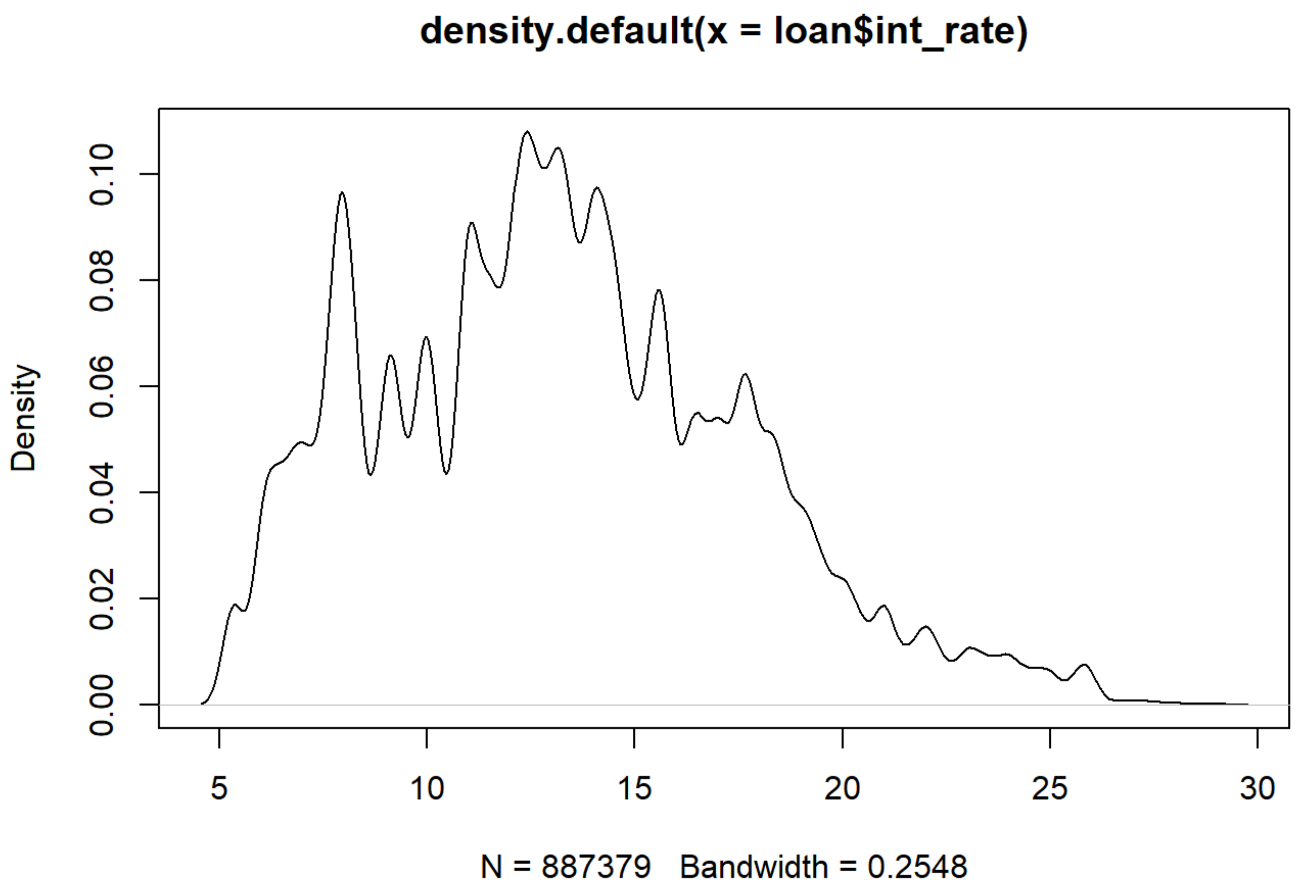
```
median(loan$int_rate)
```

```
## [1] 12.99
```

```
quantile(loan$int_rate, c(0.1, 0.25, 0.5, 0.75, 0.9))
```

```
##   10%   25%   50%   75%   90% 
##  7.69  9.99 12.99 16.20 18.99
```

```
plot(density(loan$int_rate))
```



```
# Q1 - 1.5IQR, Q1, median, Q3, Q3 + 1.5IQR, where IQR is interquartile range: Q3 -
Q1
```

reduce feature levels

```
loan$emp_length <- ifelse(loan$emp_length %in% c('1 year', '2 years', '3 years', '
4 years', '5 years', '6 years', '7 years', '8 years', '9 years'), '1~10 years', lo
an$emp_length)
head(loan$emp_length)
```

```
## [1] "10+ years"  "< 1 year"   "10+ years"  "10+ years"  "1~10 years"
## [6] "1~10 years"
```

```
table(loan$emp_length)
```

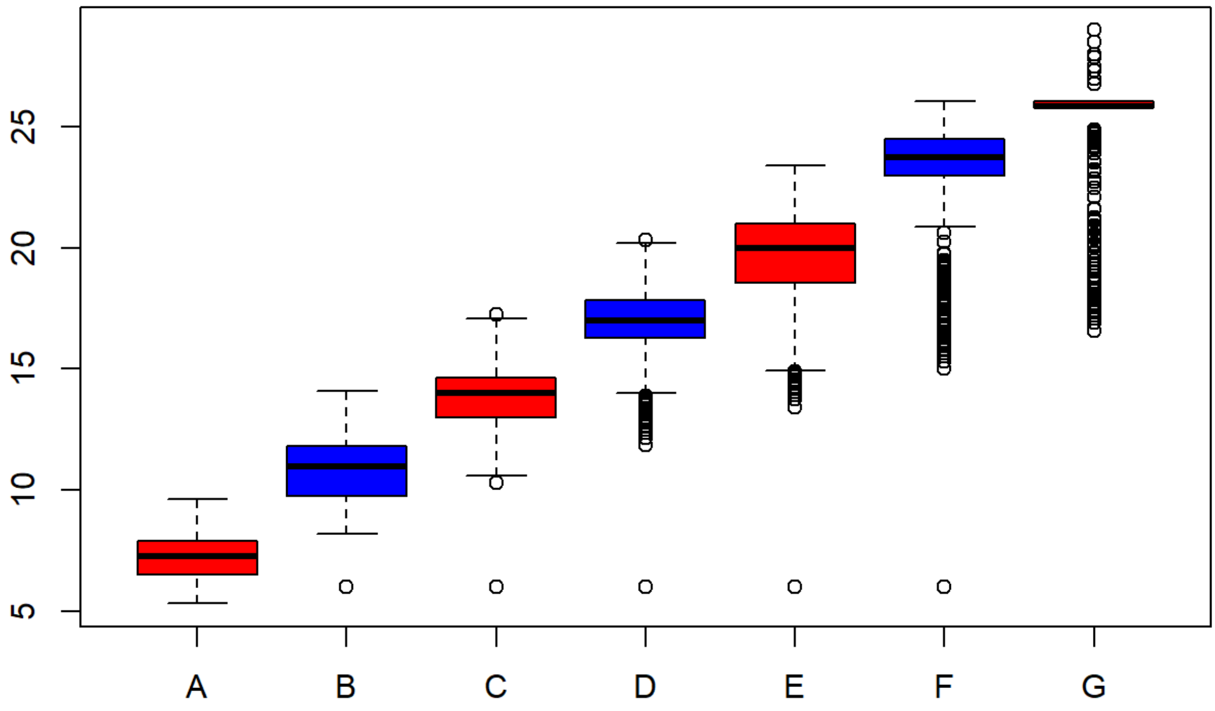
```
##
##   < 1 year 1~10 years  10+ years        n/a
##      70605     480380     291569      44825
```

generate new feature from grade

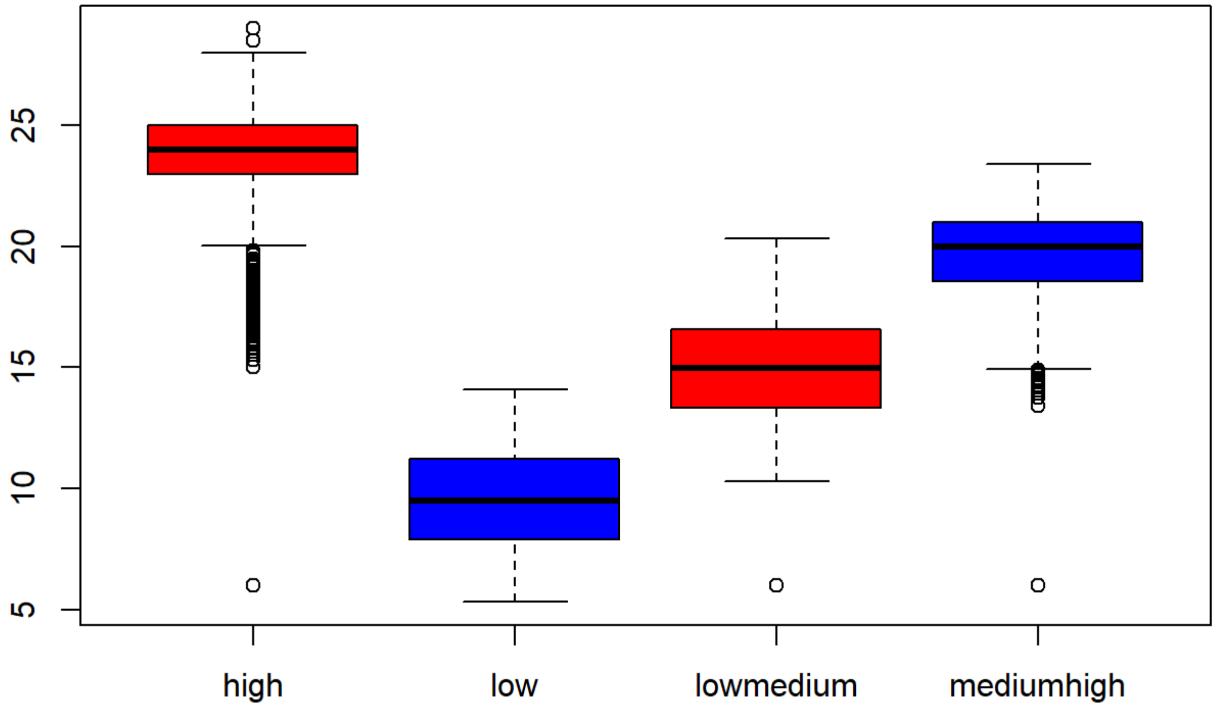
```
int_grade <- by(loan, loan$grade, function(x) {
  return(mean(x$int_rate))
})
loan$grade_mean_int <-
  ifelse(loan$grade %in% names(int_grade)[which(int_grade <= quantile(int_grade, 0.
25))],
         'low', ifelse(loan$grade %in% names(int_grade)[which(int_grade <= quantile
(int_grade, 0.5))],
                       'lowmedium', ifelse(loan$grade %in% names(int_grade)[which(
int_grade <= quantile(int_grade, 0.75))], 'mediumhigh', 'high')))
table(loan$grade_mean_int)
```

```
##
##       high        low  lowmedium mediumhigh
##      28535     402737     385402      70705
```

```
boxplot(int_rate ~ grade, data = loan, ylabel="int_rate", xlabel="grade_groups", co
l=c("red", "blue"))
```



```
boxplot(int_rate~ grade_mean_int, data = loan, ylabel="int_rate", xlabel="grade_gro
ups", col=c("red", "blue"))
```



split into training and test datasets

```
set.seed(1)
train.ind <- sample(1:dim(loan)[1], 0.7 * dim(loan)[1])
train <- loan[train.ind, ]
test <- loan[-train.ind, ]
```

linear regression model based on old features

```
mod1 <- lm(int_rate ~ grade + emp_length +annual_inc + dti +
             + term + loan_amnt + total_acc + tot_cur_bal, data = train)
summary(mod1)
```

```
##
## Call:
## lm(formula = int_rate ~ grade + emp_length + annual_inc + dti +
##     +term + loan_amnt + total_acc + tot_cur_bal, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.9086  -0.9404   0.0250   0.7490   6.0772
##
## Coefficients:
##                        Estimate Std. Error  t value Pr(>|t|)
## (Intercept)           7.256e+00  8.462e-03  857.474  < 2e-16 ***
## gradeB                3.531e+00  5.253e-03  672.307  < 2e-16 ***
## gradeC                6.724e+00  5.424e-03 1239.529  < 2e-16 ***
## gradeD                9.962e+00  6.224e-03 1600.630  < 2e-16 ***
## gradeE                1.274e+01  7.759e-03 1641.304  < 2e-16 ***
## gradeF                1.659e+01  1.164e-02 1425.514  < 2e-16 ***
## gradeG                1.893e+01  2.244e-02  843.563  < 2e-16 ***
## emp_length1~10 years  6.780e-02  6.405e-03   10.586  < 2e-16 ***
## emp_length10+ years   5.954e-02  6.663e-03    8.935  < 2e-16 ***
## emp_lengthn/a        -1.265e-02  9.452e-03   -1.338    0.181
## annual_inc           -4.716e-07  2.851e-08  -16.544  < 2e-16 ***
## dti                  -6.320e-04  8.216e-05   -7.691 1.46e-14 ***
## term 60 months        4.133e-02  4.422e-03    9.345  < 2e-16 ***
## loan_amnt             2.194e-06  2.369e-07    9.262  < 2e-16 ***
## total_acc            -1.451e-03  1.503e-04   -9.654  < 2e-16 ***
## tot_cur_bal          -1.941e-07  1.265e-08  -15.348  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.26 on 572045 degrees of freedom
##   (49104 observations deleted due to missingness)
## Multiple R-squared:  0.9183, Adjusted R-squared:  0.9183
## F-statistic: 4.285e+05 on 15 and 572045 DF,  p-value: < 2.2e-16
```

linear regression model based on new features

```
mod2 <- lm(int_rate ~ grade_mean_int + emp_length + annual_inc + dti +
             + term + loan_amnt + total_acc + tot_cur_bal, data = train)
summary(mod2)
```

```
##
## Call:
## lm(formula = int_rate ~ grade_mean_int + emp_length + annual_inc +
##     dti + +term + loan_amnt + total_acc + tot_cur_bal, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.487  -1.573  -0.148   1.537  12.352
##
## Coefficients:
##                             Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                2.403e+01  1.904e-02 1261.736  < 2e-16 ***
## grade_mean_intlow         -1.434e+01  1.584e-02 -905.467  < 2e-16 ***
## grade_mean_intlowmedium   -8.896e+00  1.542e-02 -577.057  < 2e-16 ***
## grade_mean_intmediumhigh  -4.217e+00  1.732e-02 -243.502  < 2e-16 ***
## emp_length1~10 years       8.375e-02  1.003e-02    8.347  < 2e-16 ***
## emp_length10+ years        5.877e-02  1.044e-02    5.630 1.81e-08 ***
## emp_lengthn/a              5.951e-02  1.481e-02    4.019 5.85e-05 ***
## annual_inc                -1.227e-06  4.464e-08  -27.482  < 2e-16 ***
## dti                        1.540e-03  1.287e-04   11.970  < 2e-16 ***
## term 60 months             6.218e-01  6.854e-03   90.729  < 2e-16 ***
## loan_amnt                 -1.193e-06  3.708e-07   -3.216   0.0013 **
## total_acc                 -4.889e-03  2.355e-04  -20.765  < 2e-16 ***
## tot_cur_bal               -7.743e-07  1.979e-08  -39.126  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.975 on 572048 degrees of freedom
##   (49104 observations deleted due to missingness)
## Multiple R-squared:  0.7994, Adjusted R-squared:  0.7994
## F-statistic: 1.9e+05 on 12 and 572048 DF,  p-value: < 2.2e-16
```

identify NA in feature of model

```
train.sub <- train[, c('int_rate', 'grade_mean_int', 'emp_length', 'annual_inc', 'dti', 'term', 'loan_amnt', 'total_acc', 'tot_cur_bal')]
dim(train.sub)
```

```
## [1] 621165      9
```

```
num.NA <- sort(sapply(train.sub, function(x) { sum(is.na(x))} ), decreasing = TRUE)
num.NA
```

```
##    tot_cur_bal      total_acc     annual_inc       int_rate grade_mean_int
##          49104             18              2              0              0
##     emp_length            dti           term      loan_amnt
##              0              0              0              0
```

imputate NA with median of each feature

```
train.sub$tot_cur_bal[which(is.na(train.sub$tot_cur_bal))] <- median(train.sub$tot_
cur_bal, na.rm = T)
train.sub$total_acc[which(is.na(train.sub$total_acc))] <- median(train.sub$total_ac
c, na.rm = T)
train.sub$annual_inc[which(is.na(train.sub$annual_inc))] <- median(train.sub$annual
_inc, na.rm = T)
num.NA <- sort(sapply(train.sub, function(x) { sum(is.na(x))} ), decreasing = TRUE)
num.NA
```

```
##       int_rate grade_mean_int     emp_length     annual_inc            dti
##              0              0              0              0              0
##           term      loan_amnt      total_acc    tot_cur_bal
##              0              0              0              0
```

linear regression of mod2 without NA deletion

```
mod2 <- lm(int_rate ~ ., data = train.sub)
summary(mod2)
```

```
##
## Call:
## lm(formula = int_rate ~ ., data = train.sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.6470  -1.5875  -0.1268   1.5465  12.7183
##
## Coefficients:
##                             Estimate Std. Error  t value Pr(>|t|)
## (Intercept)               2.360e+01  1.832e-02 1288.139  < 2e-16 ***
## grade_mean_intlow        -1.397e+01  1.527e-02 -914.799  < 2e-16 ***
## grade_mean_intlowmedium  -8.547e+00  1.488e-02 -574.270  < 2e-16 ***
## grade_mean_intmediumhigh -3.994e+00  1.674e-02 -238.537  < 2e-16 ***
## emp_length1~10 years      1.001e-01  9.612e-03   10.417  < 2e-16 ***
## emp_length10+ years       8.519e-02  1.005e-02    8.479  < 2e-16 ***
## emp_lengthn/a             8.800e-02  1.444e-02    6.095 1.09e-09 ***
## annual_inc               -1.281e-06  4.346e-08  -29.474  < 2e-16 ***
## dti                       2.121e-03  1.294e-04   16.400  < 2e-16 ***
## term 60 months            6.419e-01  6.663e-03   96.343  < 2e-16 ***
## loan_amnt                 2.546e-06  3.603e-07    7.067 1.59e-12 ***
## total_acc                -5.060e-03  2.288e-04  -22.111  < 2e-16 ***
## tot_cur_bal              -8.197e-07  1.971e-08  -41.578  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.996 on 621152 degrees of freedom
## Multiple R-squared:  0.7925, Adjusted R-squared:  0.7925
## F-statistic: 1.977e+05 on 12 and 621152 DF,  p-value: < 2.2e-16
```

check data before scaling

```
head(train.sub)
```

```
##        int_rate grade_mean_int emp_length annual_inc   dti       term
## 235607     8.19            low  10+ years     128941 31.83  36 months
## 330215    22.15     mediumhigh 1~10 years     100000 12.82  60 months
## 508337     7.26            low  10+ years      50000 24.96  36 months
## 805922    12.29      lowmedium 1~10 years      75000 24.06  36 months
## 178968    21.98     mediumhigh 1~10 years      48000 16.70  60 months
## 797208     6.68            low  10+ years      55000 17.61  36 months
##        loan_amnt total_acc tot_cur_bal
## 235607     28000        33      530944
## 330215     14000        29      146665
## 508337      6000        23      140834
## 805922     12500        44      214499
## 178968     20000        16       17999
## 797208      7000        17      144220
```

data standardization

```
train.sub.scale <- train.sub
train.sub.scale[, c(4,5,7,8,9)] <- scale(train.sub.scale[, c(4,5,7,8,9)])
mod3 <- lm(int_rate ~ ., data = as.data.frame(train.sub.scale))
summary(mod3)
```

```
##
## Call:
## lm(formula = int_rate ~ ., data = as.data.frame(train.sub.scale))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.6470  -1.5875  -0.1268   1.5465  12.7183
##
## Coefficients:
##                           Estimate Std. Error  t value Pr(>|t|)
## (Intercept)              23.337603   0.017274 1351.006  < 2e-16 ***
## grade_mean_intlow       -13.966351   0.015267 -914.799  < 2e-16 ***
## grade_mean_intlowmedium  -8.547058   0.014883 -574.270  < 2e-16 ***
## grade_mean_intmediumhigh  -3.994140   0.016744 -238.537  < 2e-16 ***
## emp_length1~10 years       0.100128   0.009612   10.417  < 2e-16 ***
## emp_length10+ years        0.085194   0.010048    8.479  < 2e-16 ***
## emp_lengthn/a              0.088003   0.014438    6.095 1.09e-09 ***
## annual_inc               -0.084471   0.002866  -29.474  < 2e-16 ***
## dti                       0.042032   0.002563   16.400  < 2e-16 ***
## term 60 months            0.641948   0.006663   96.343  < 2e-16 ***
## loan_amnt                 0.021473   0.003039    7.067 1.59e-12 ***
## total_acc                -0.059961   0.002712  -22.111  < 2e-16 ***
## tot_cur_bal              -0.121456   0.002921  -41.578  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.996 on 621152 degrees of freedom
## Multiple R-squared:  0.7925, Adjusted R-squared:  0.7925
## F-statistic: 1.977e+05 on 12 and 621152 DF,  p-value: < 2.2e-16
```