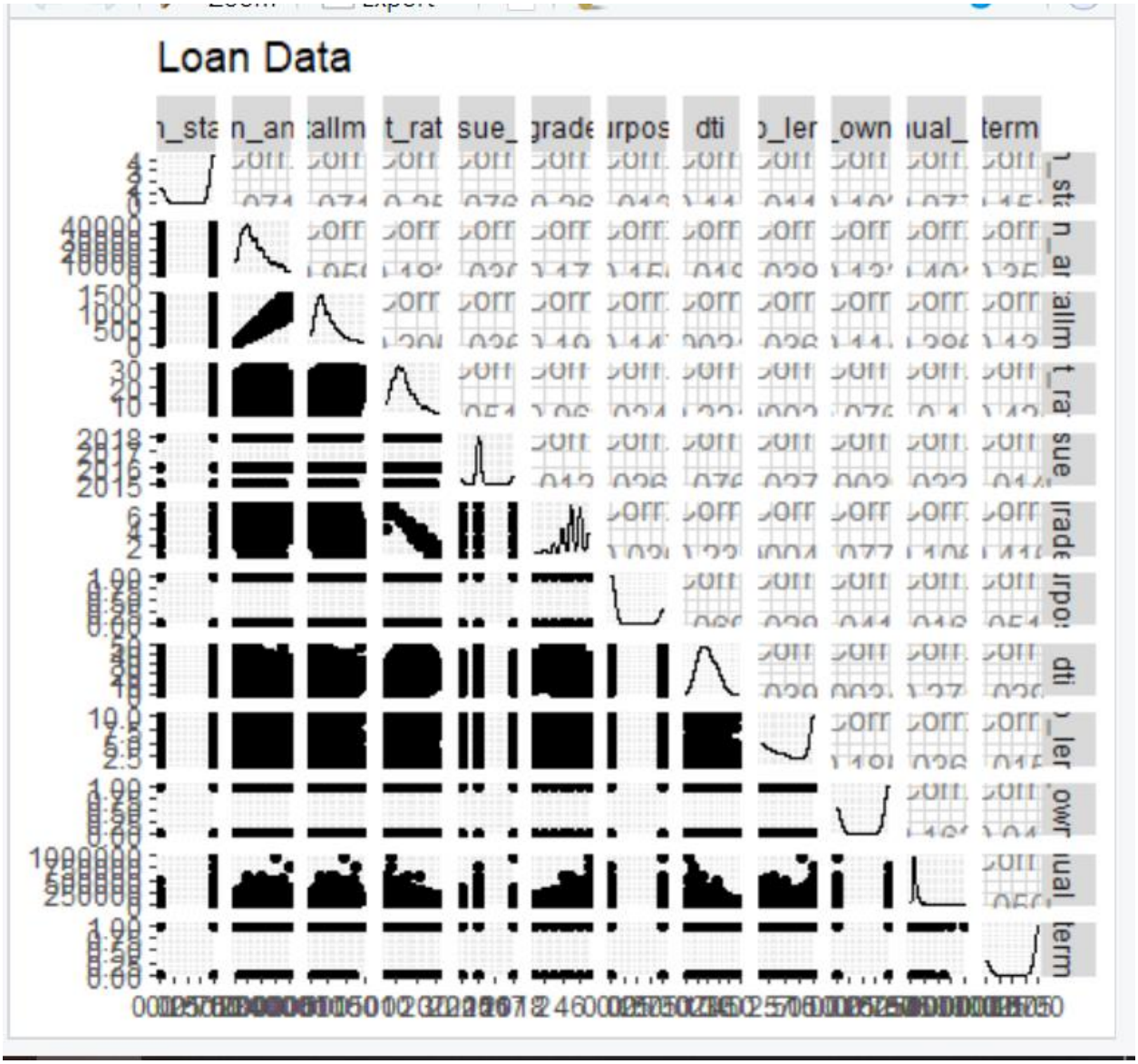
```
############ Multinomial Linear Regression ########################

View(loan)

mysample = ABC

View(ABC)

mysample = loan[,c("loan_status","loan_amnt" ,"installment", "int_rate","issue_d","grade","purpose",
"dti",

                "emp_length" , "home_ownership" ,"annual_inc" , "term")]

mysample <- mysample[sample(1:nrow(mysample), 5000,replace=FALSE),]




#Multiple Regression

View(mysample)


# Performing multiple regression on DEF dataset

fit <- lm(loan_status~loan_amnt+installment+int_rate+

        issue_d+grade+purpose+dti+emp_length+home_ownership+

        annual_inc+term, data = mysample)

#show the results

summary(fit)


#ANS- By looking at the output we can figure out that int_rate and emp_length are insignificant


#Summary has three sections. Section1: How well does the model fit the data (before Coefficients).
Section2: Is the hypothesis supported? (until sifnif codes). Section3: How well does data fit the model
(again).

# Useful Helper Functions

coefficients(fit)

library(ggplot2)

install.packages("GGally")
```

```
library(GGally)

install.packages("tidyverse")

library(tidyverse)

# install.packages("rlang")

install.packages("https://cran.r-project.org/src/contrib/Archive/rlang/rlang_0.4.4.tar.gz", repo=NULL, type = "source")

install.packages("caret")

library(caret)

ggpairs(data=mysample, title="Loan Data")

#install.packages("GGally", lib="/Library/Frameworks/R.framework/Versions/3.5/Resources/library")

library(GGally)

# ggpairs(data=mtcars, title="Cars Data")

confint(fit,level=0.95)

# Predicted Values

fitted(fit)

residuals(fit)

#Anova Table

anova(fit)


#Anova fit tells me that installment, purpose, emp_length are non-significant



#dont worry for next two lines

vcov(fit)

cov2cor(vcov(fit))
```



#acting as outliers for your dataset

temp <- influence.measures(fit)

temp

```
Influence measures of
        lm(formula = loan_status ~ loan_amnt + installment + int_rate +      issue_d + grade + purpose + dti + emp_length + home_ownership +
     annual_inc + term, data = mysample) :

         dfb.1_  dfb.ln_m  dfb.inst  dfb.int_  dfb.iss_  dfb.grad  dfb.prps
68791  -8.63e-03 -3.71e-02  0.035299 -0.011604  8.68e-03 -0.006422  0.010377
123595  2.08e-02  1.29e-02 -0.012804 -0.006883 -2.07e-02 -0.011505  0.023278
70152  -8.85e-06 -3.98e-05 -0.000961 -0.007688  8.36e-05 -0.009362 -0.005107
130291  1.45e-02 -6.45e-03  0.005157 -0.012461 -1.44e-02 -0.007788  0.019424
20405  -7.96e-03 -7.58e-02  0.072162 -0.013996  8.07e-03  0.000153  0.005951
35175   2.19e-03 -3.62e-04  0.000490  0.004408 -2.22e-03  0.006160 -0.001689
55469   4.20e-03 -8.23e-03  0.011909  0.008211 -4.29e-03  0.009547  0.035053
11511   7.33e-02 -3.63e-04 -0.001713  0.018369 -7.33e-02  0.017561 -0.051176
14566  -1.36e-04 -2.13e-02  0.020255 -0.010126  2.47e-04 -0.001508 -0.003903
39474   3.16e-03 -8.43e-04  0.001808  0.003327 -3.20e-03  0.005381  0.015323
88739  -1.31e-02 -6.21e-03  0.002575 -0.023137  1.33e-02 -0.035648  0.012739
82418   3.20e-04 -1.11e-03  0.000679 -0.002892 -2.85e-04 -0.002602 -0.002689
36316   1.09e-03 -7.62e-03  0.006824 -0.008676 -1.01e-03 -0.004990 -0.004930
11355  -1.52e-02  1.54e-03 -0.003832  0.003512  1.51e-02 -0.003999 -0.008778
124172 -3.66e-02 -2.71e-03 -0.000419  0.027101  3.63e-02  0.022490  0.009978
7461   -6.12e-03 -1.13e-03  0.001047  0.000754  6.09e-03  0.002284 -0.001061
51998  -1.50e-02 -2.32e-02  0.024897 -0.046336  1.54e-02 -0.030391  0.019273
108008 -1.48e-03 -4.26e-03  0.007435 -0.011192  1.56e-03 -0.009852 -0.001483
13909   5.41e-02 -3.60e-02  0.031003 -0.022977 -5.38e-02 -0.014680  0.005726
81769   1.60e-03 -3.06e-05 -0.000367  0.002199 -1.63e-03  0.003127 -0.003308
25214   5.36e-03  1.21e-02 -0.011331  0.011164 -5.45e-03  0.009550 -0.004089
26877  -1.09e-02 -6.58e-03  0.006609 -0.023159  1.10e-02 -0.026499  0.009997
123870 -4.08e-02 -1.27e-02  0.015612 -0.011341  4.08e-02 -0.005525  0.014854
89770   2.03e-03 -2.91e-03  0.003664  0.023508 -2.22e-03  0.029036 -0.039330
1154   -2.11e-02  2.79e-03 -0.003257 -0.003190  2.11e-02 -0.006677 -0.005634
```

#Anova Table

anova(fit)

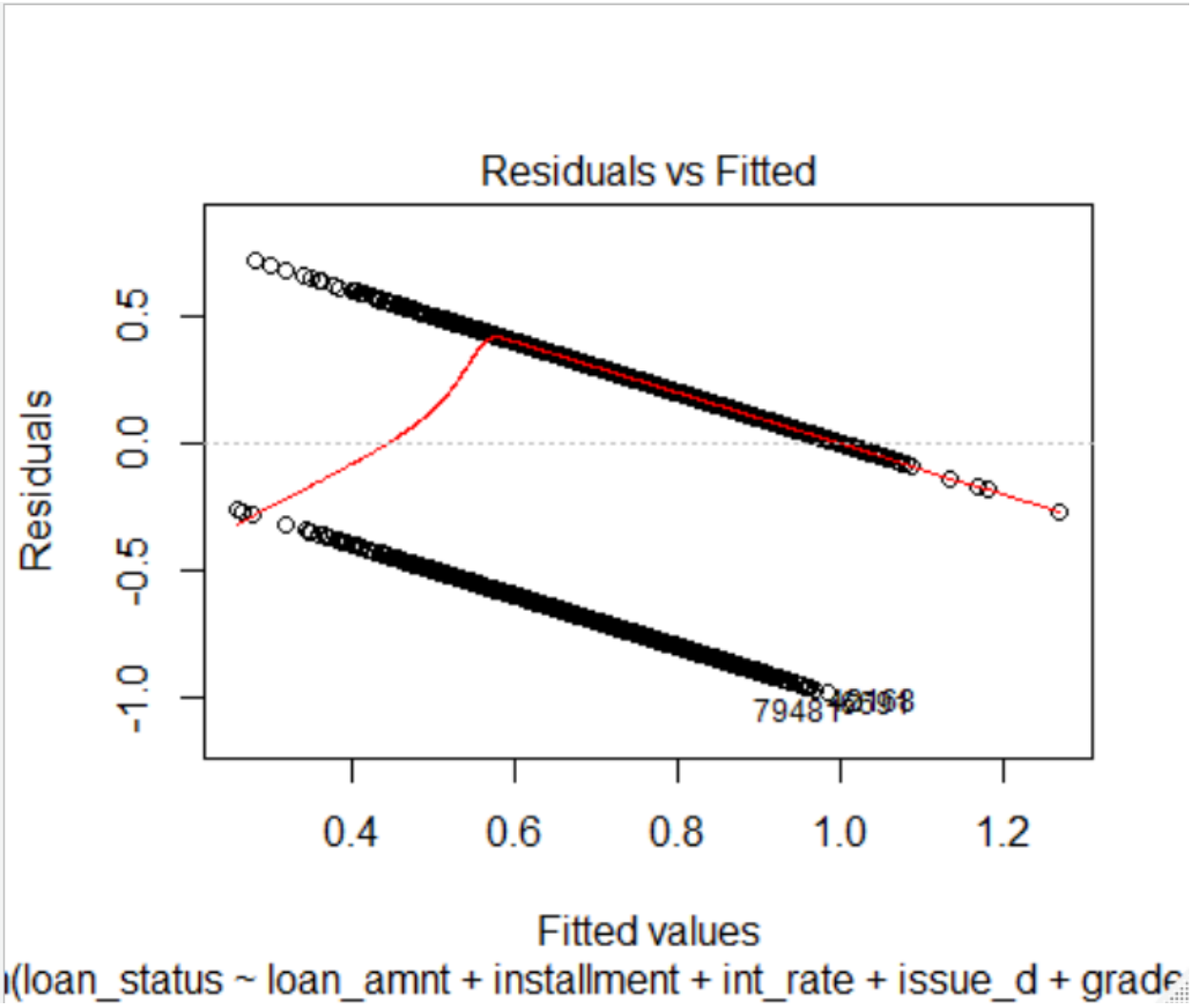
#Anova fit tells me that installment, purpose, emp_length are non-significant

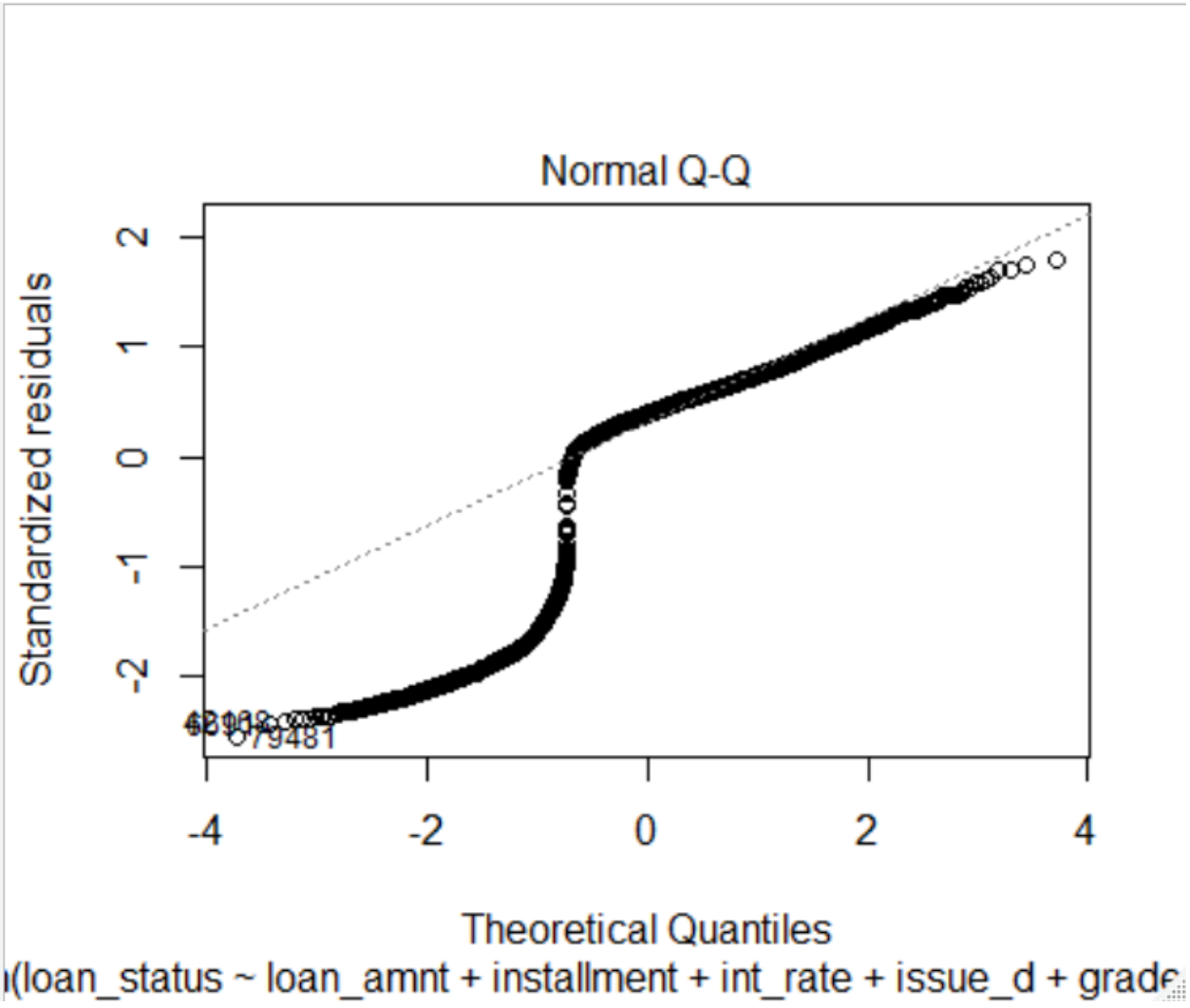
```
Analysis of Variance Table

Response: loan_status
                  Df Sum Sq Mean Sq  F value    Pr(>F)
loan_amnt          1   4.59   4.594  28.1988 1.142e-07 ***
installment        1   0.10   0.098   0.6044  0.436945
int_rate           1  52.11  52.108 319.8468 < 2.2e-16 ***
issue_d            1   7.05   7.055  43.3031 5.170e-11 ***
grade              1   2.52   2.523  15.4866 8.422e-05 ***
purpose            1   0.09   0.095   0.5802  0.446280
dti                1   2.79   2.787  17.1076 3.590e-05 ***
emp_length         1   0.15   0.151   0.9241  0.336442
home_ownership     1   7.17   7.174  44.0329 3.572e-11 ***
annual_inc         1   1.69   1.687  10.3581  0.001297 **
term               1   3.73   3.732  22.9075 1.749e-06 ***
Residuals       4988 812.62   0.163
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

#diagnostic plots

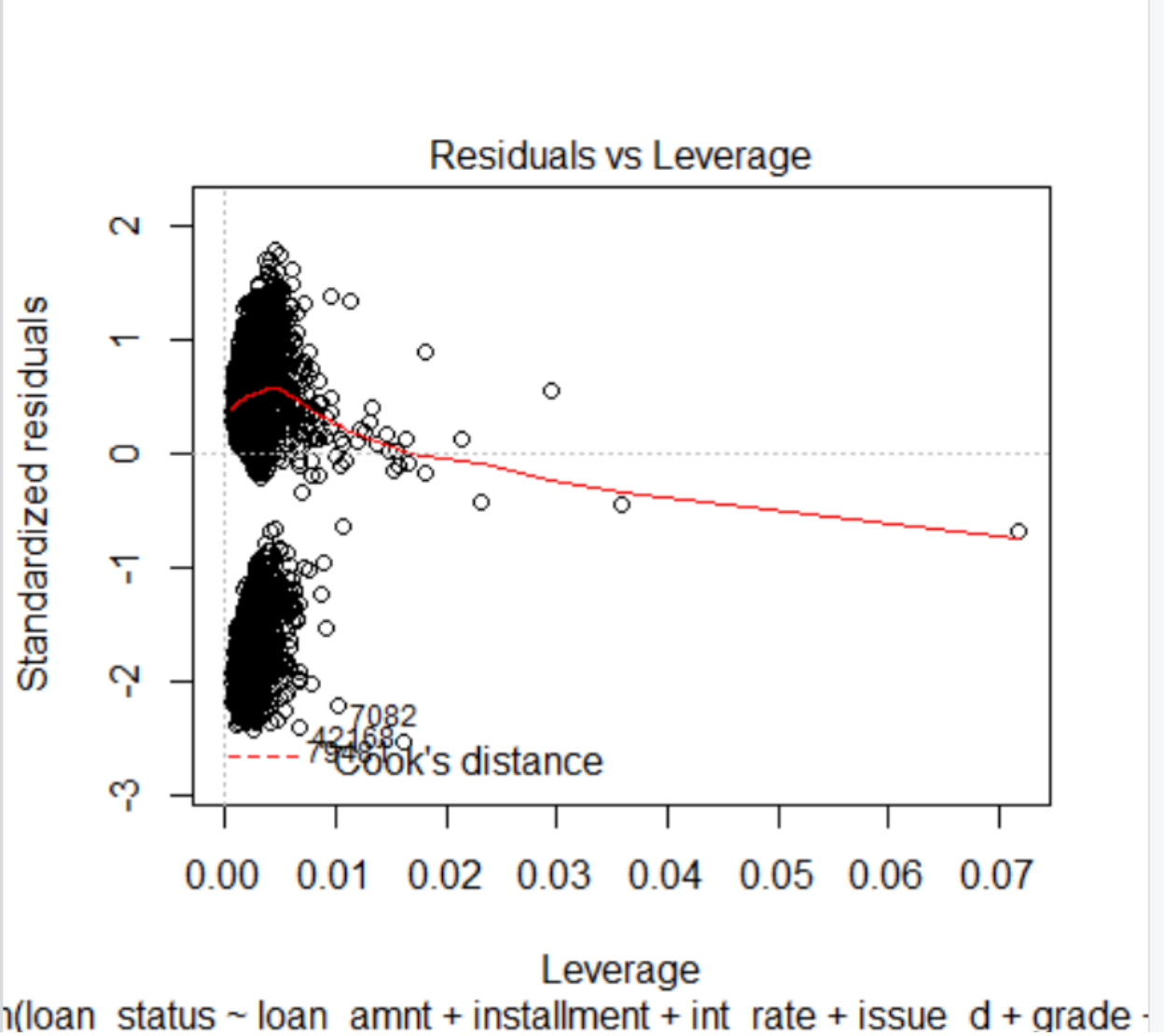
Diagnostic plots provide checks for heteroscedasticity, normality, and influential observations. The following code provides a simultaneous test that the below five variables we chose adds to linear prediction:

plot(fit)


Residuals vs Fitted
Residuals
0.5
0.0
-0.5
-1.0
0.4
0.6
0.8
1.0
1.2
Fitted values
(loan_status ~ loan_amnt + installment + int_rate + issue_d + grade



(loan_status ~ loan_amnt + installment + int_rate + issue_d + grade



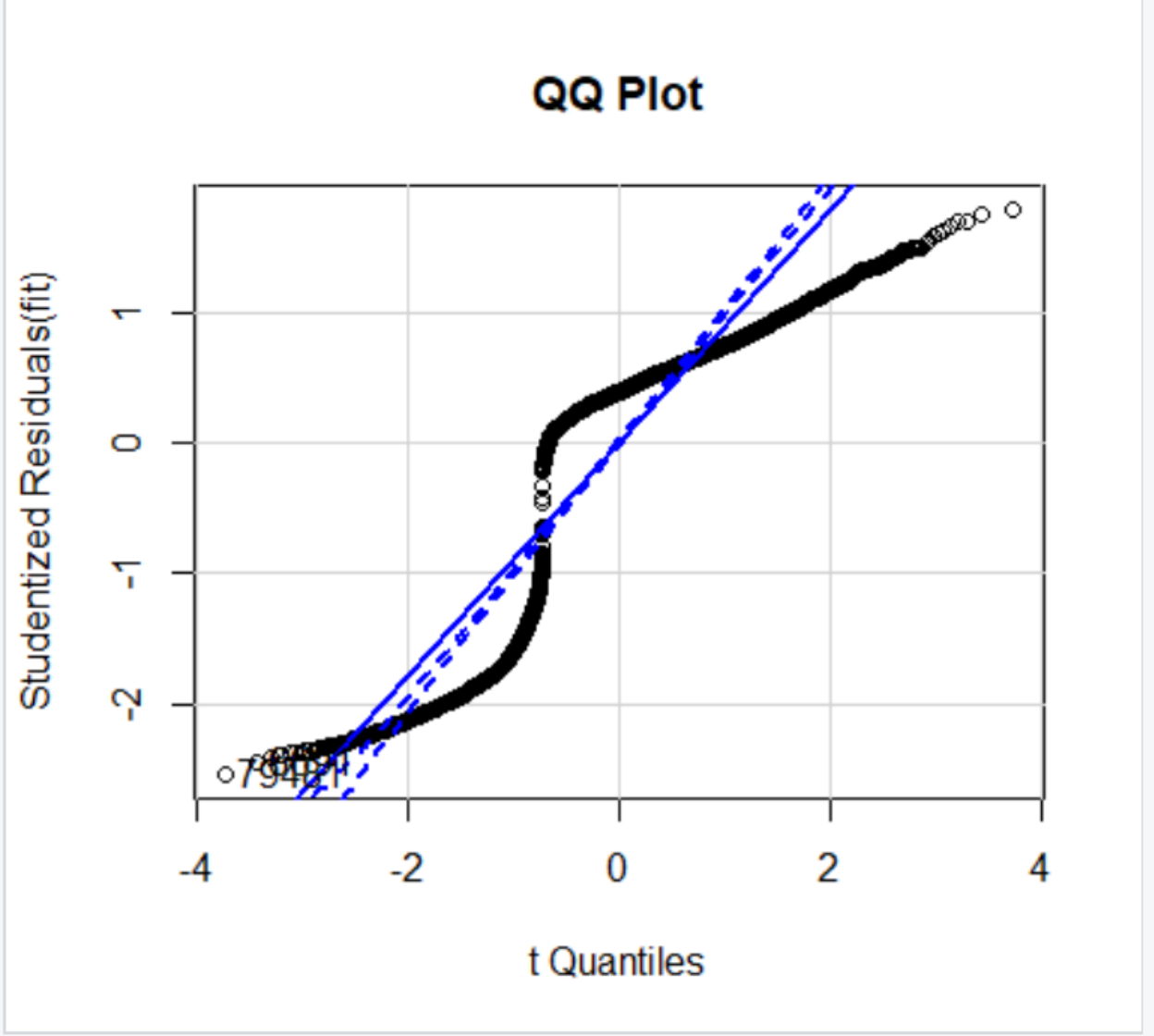
Assessing Outliers

```
library(car)
```

```
outlierTest(fit)
```

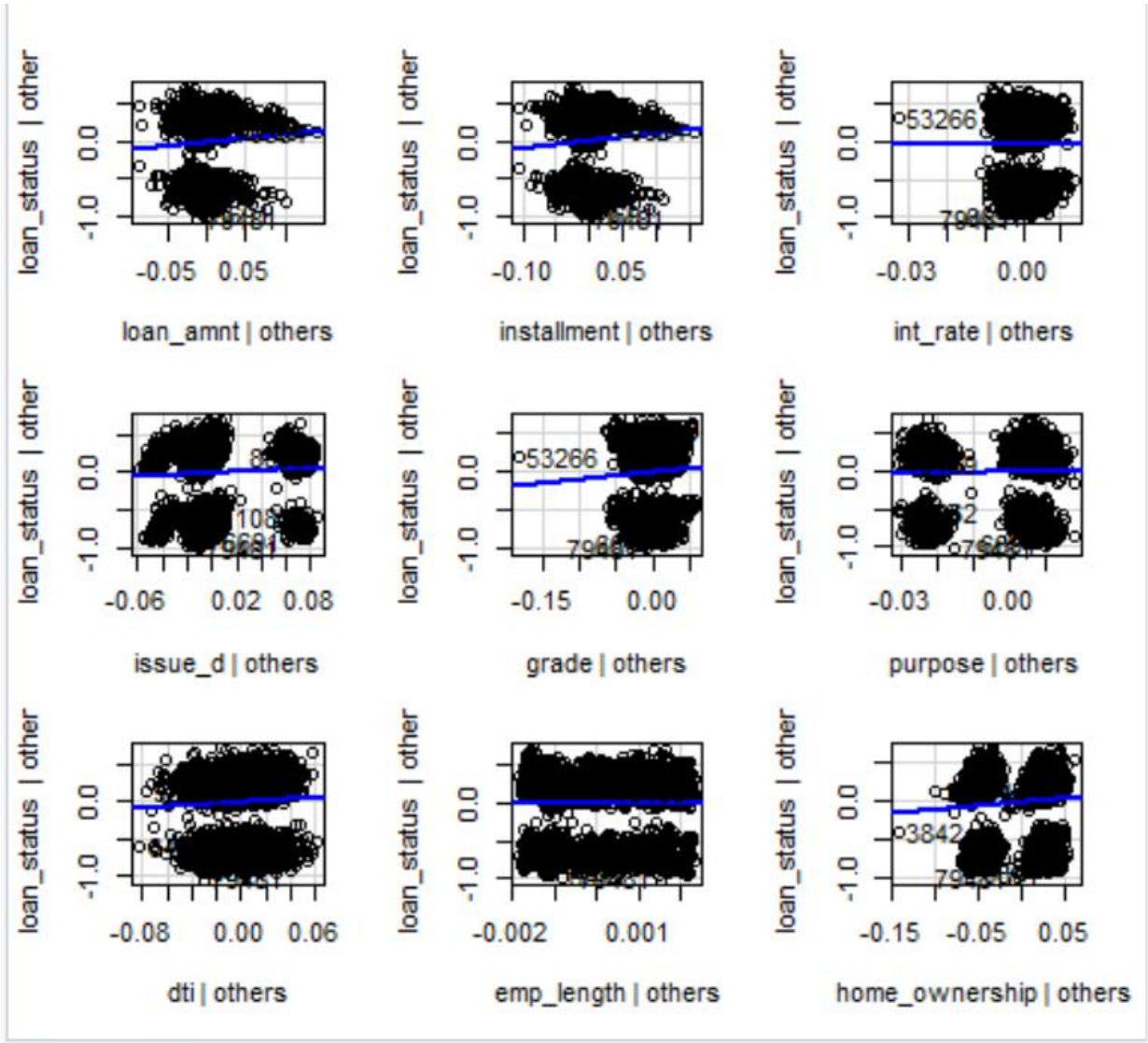
```
No Studentized residuals with Bonferroni p < 0.05
Largest |rstudent|:
       rstudent unadjusted p-value Bonferroni p
79481 -2.540769           0.011091           NA
```

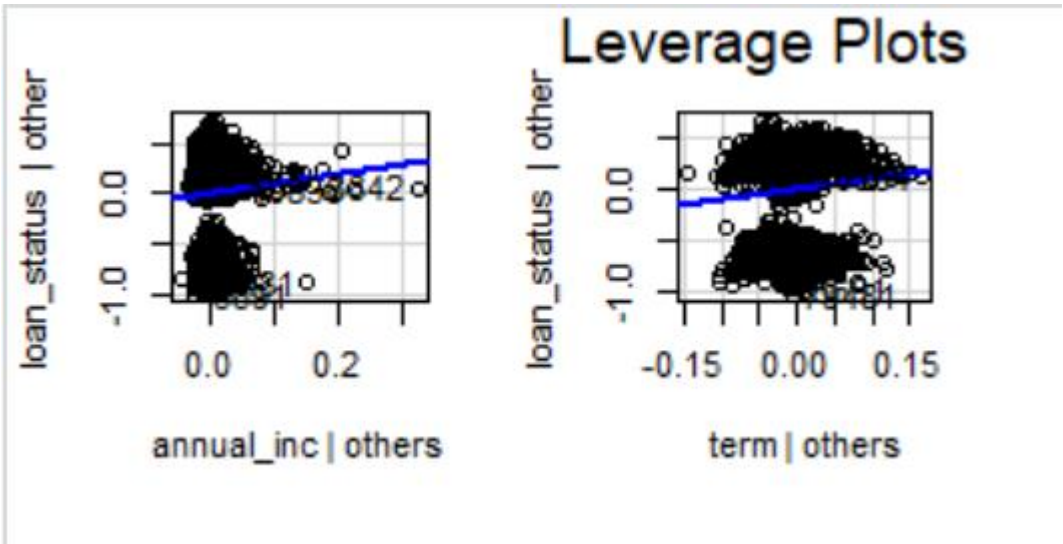
```
qqPlot(fit, main="QQ Plot")
```




```
6691 79481
2024  3279
```

leveragePlots(fit) # leverage plots

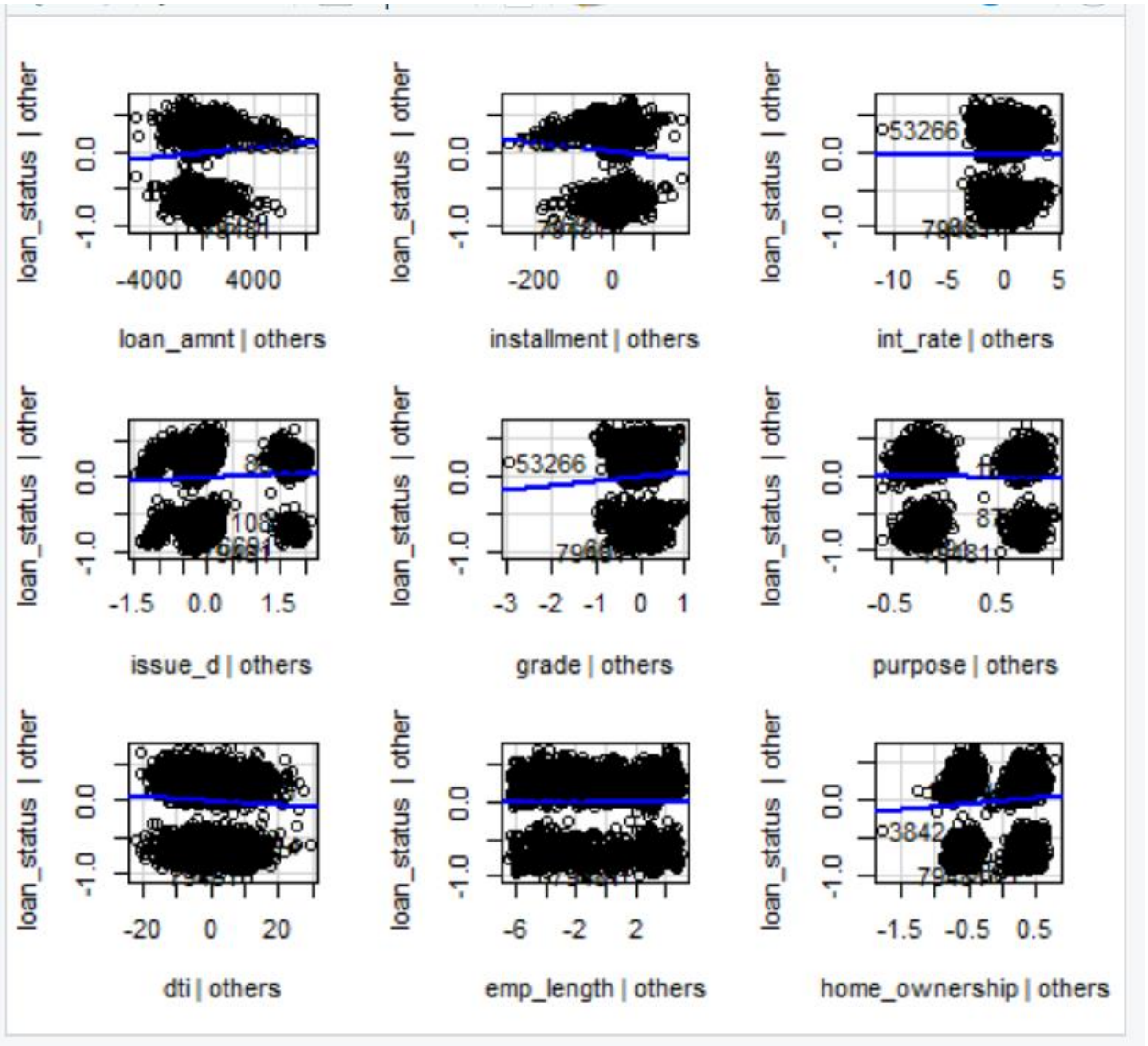

loan_status | other
loan_amnt | others
installment | others
int_rate | others
issue_d | others
grade | others
purpose | others
dti | others
emp_length | others
home_ownership | others

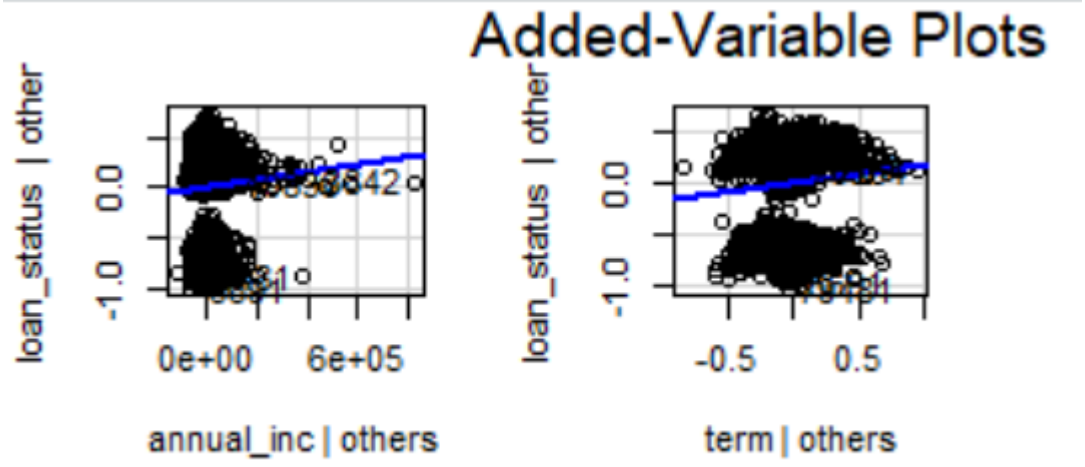

Leverage Plots
loan_status | other
annual_inc | others
term | others

Influential Observations

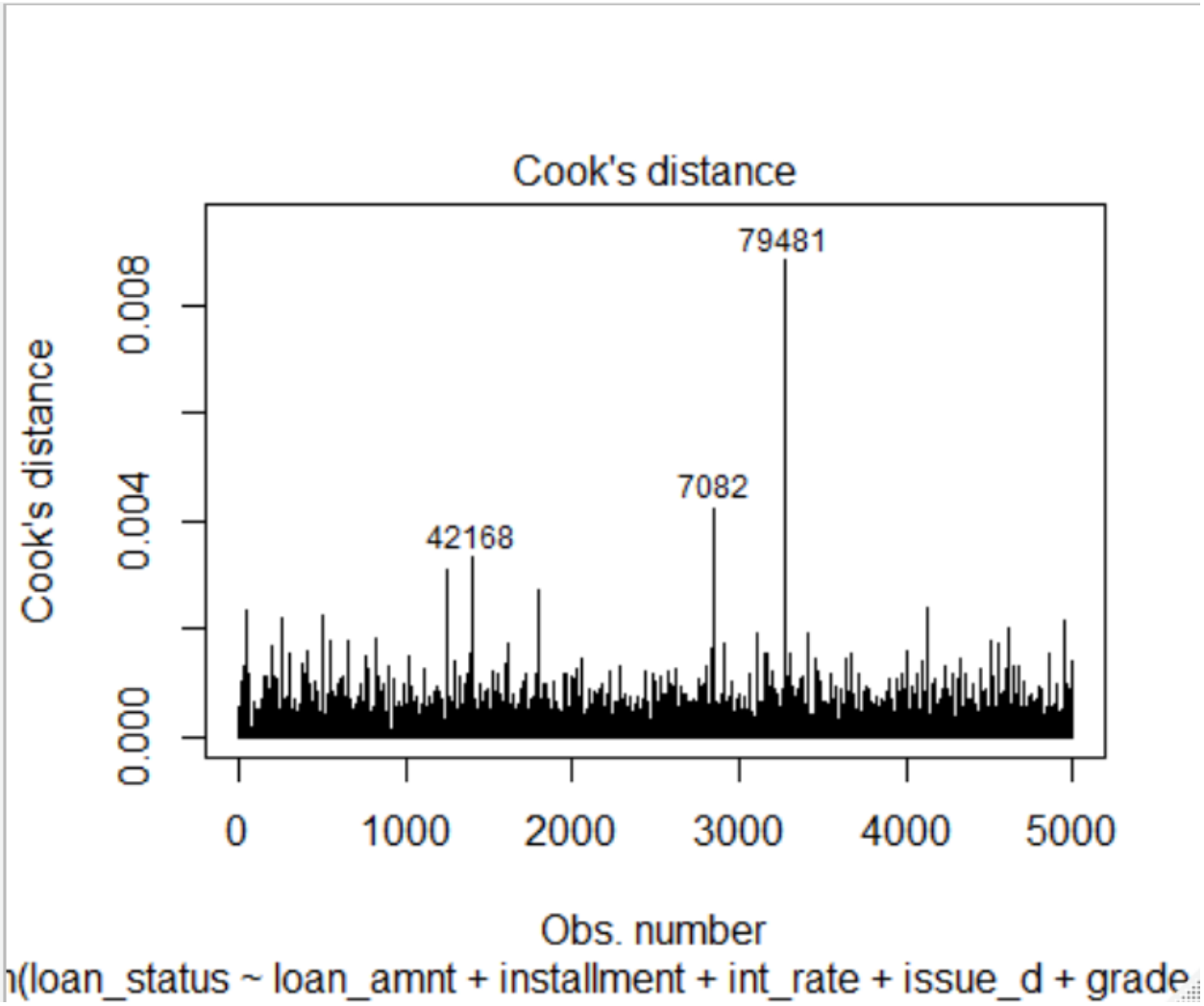
added variable plots

avPlots(fit)



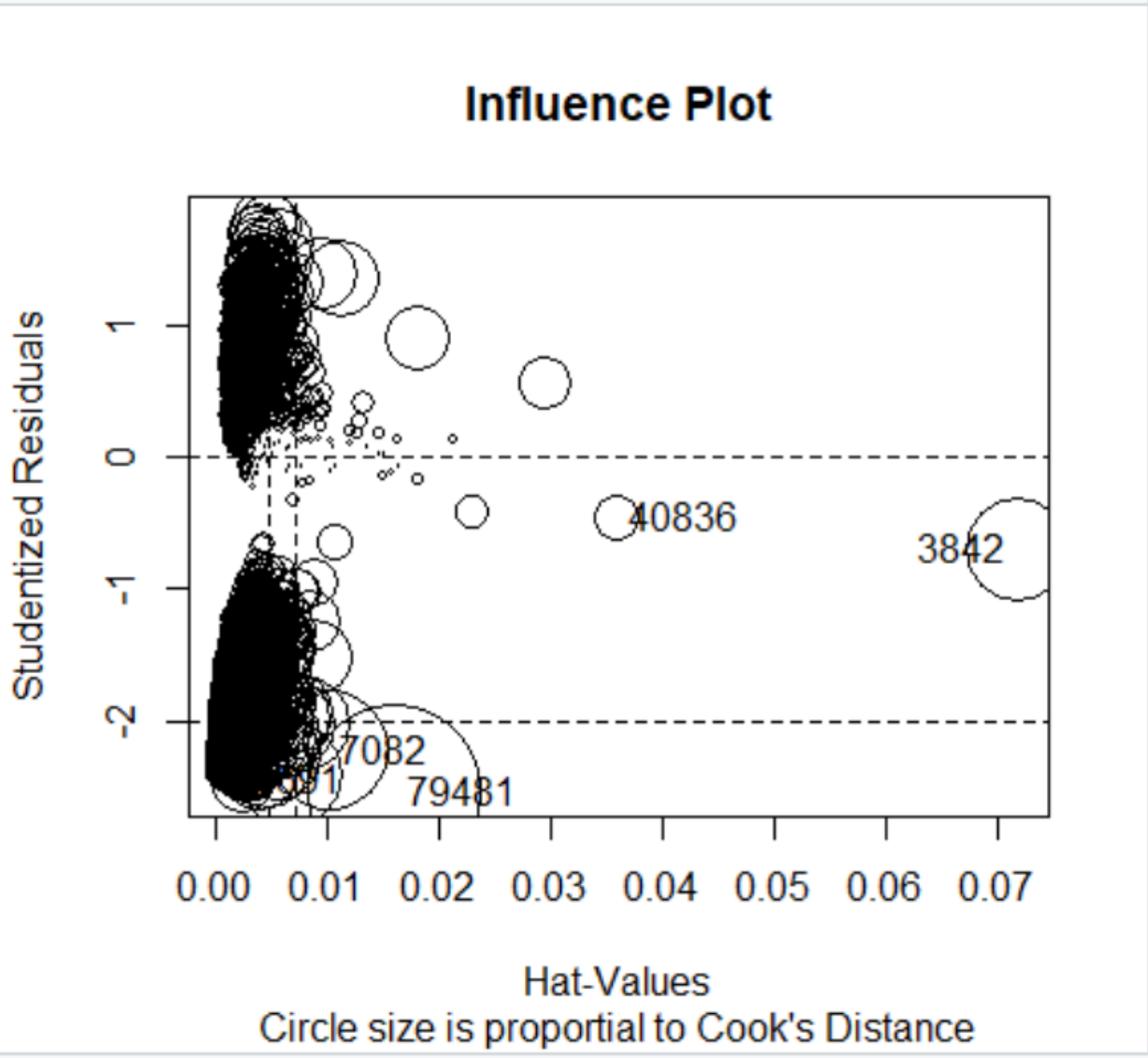



```
# Cook's D plot
# identify D values > 4/(n-k-1)
cutoff <- 4/((nrow(mtcars)-length(fit$coefficients)-2))
plot(fit, which=4, cook.levels=cutoff)
```



Influence Plot

influencePlot(fit, id.method="identify", main="Influence Plot", sub="Circle size is proportial to Cook's Distance")

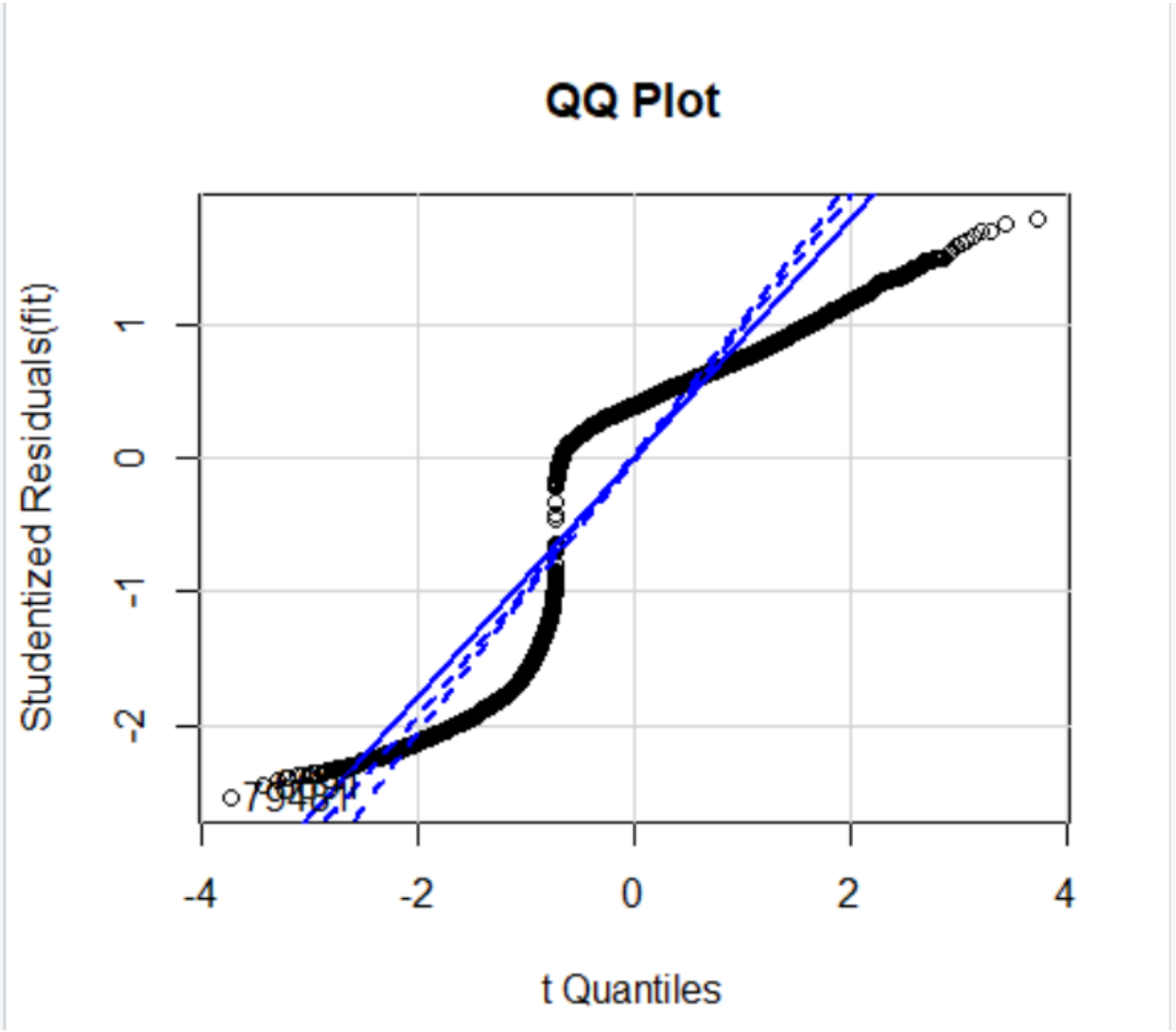


	StudRes	Hat	CookD
3842	-0.6927814	0.071812054	0.0030946957
40836	-0.4537145	0.035918266	0.0006392264
6691	-2.4433789	0.002472099	0.0012317104
7082	-2.2209780	0.010192219	0.0042294398
79481	-2.5407687	0.016162472	0.0088279261

Normality of Residuals

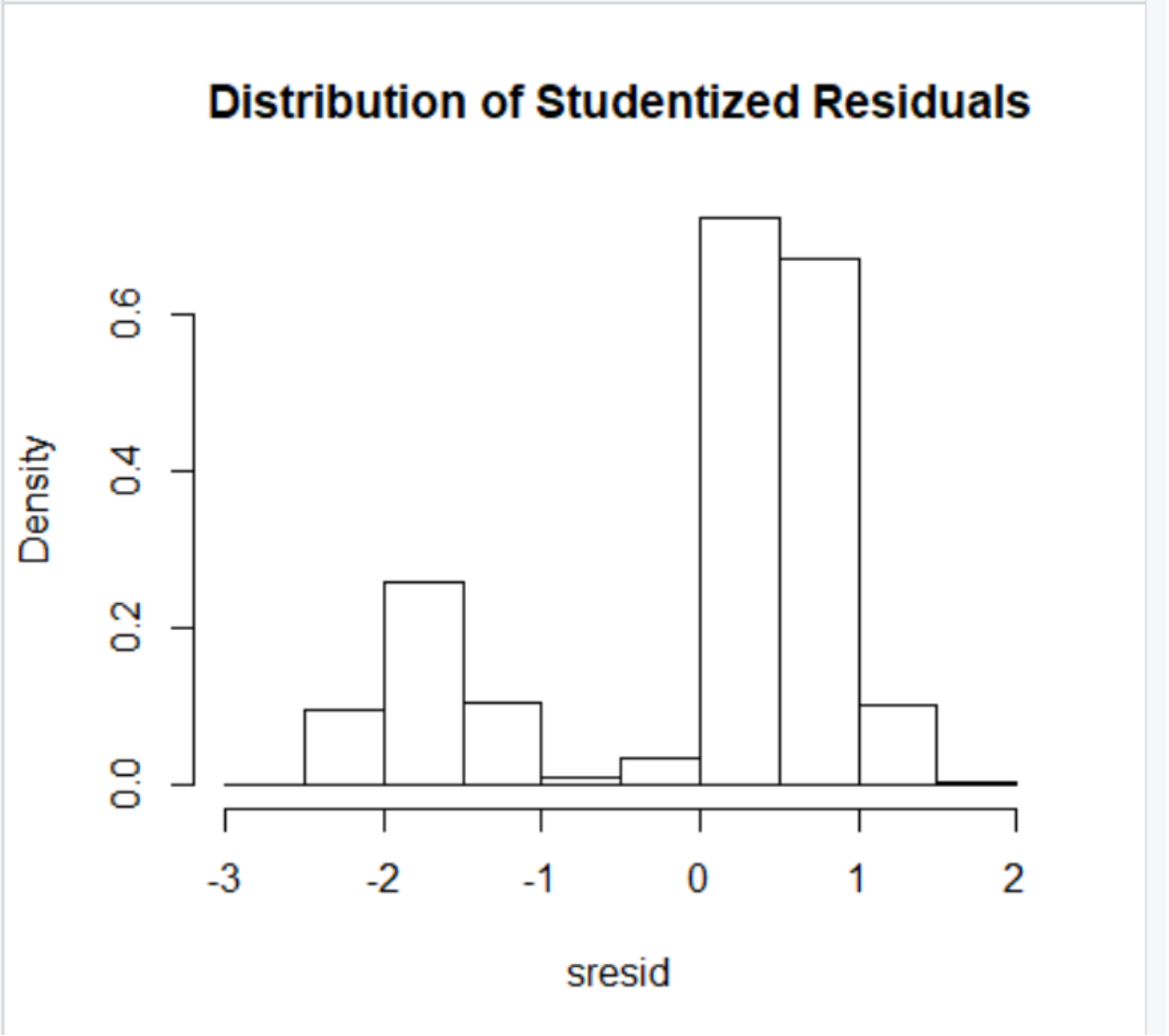
qq plot for studentized resid

```
qqPlot(fit, main="QQ Plot")
```

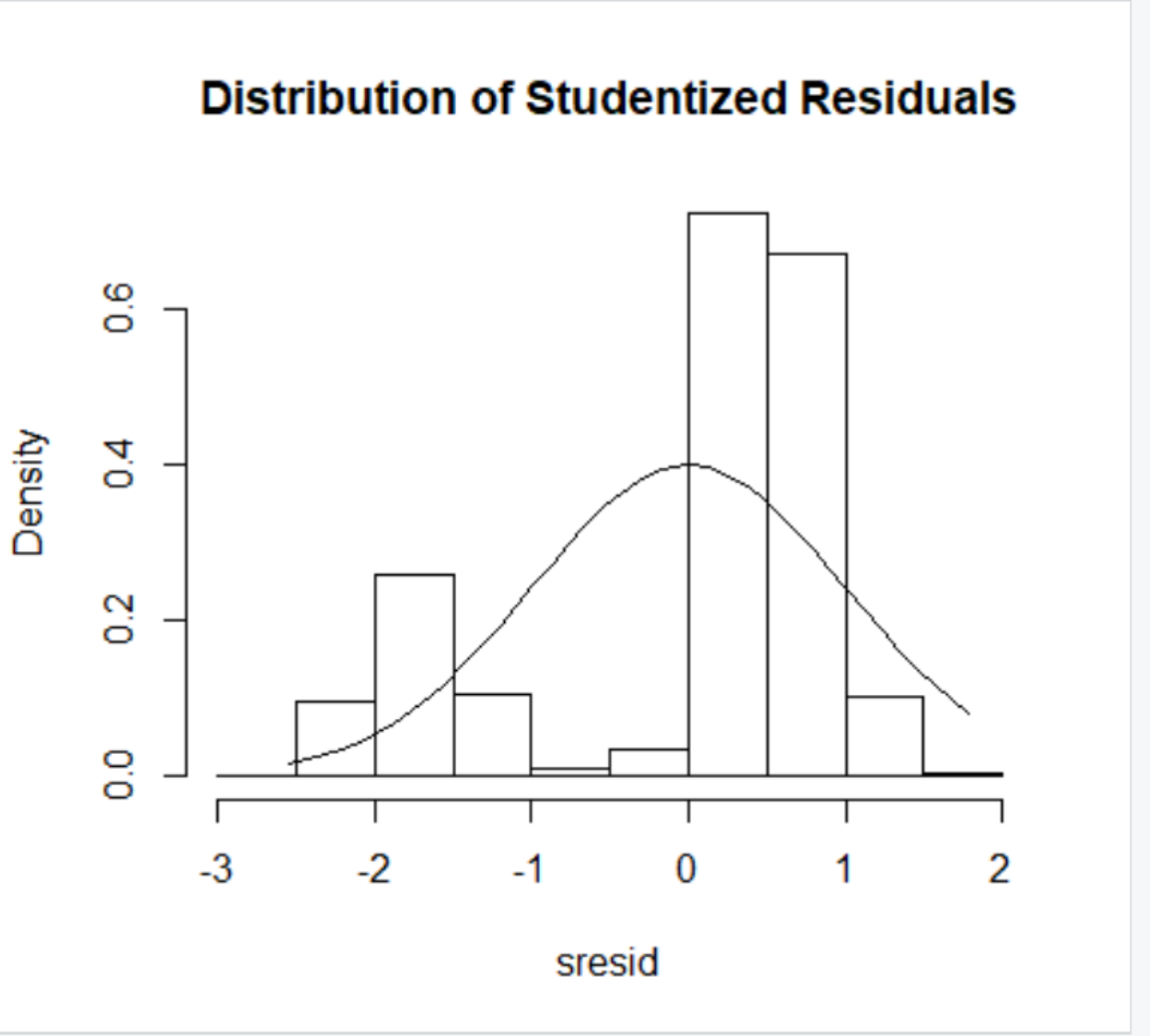



```
6691 79481
2024  3279
```

```
# distribution of studentized residuals
library(MASS)
sresid <- studres(fit)
hist(sresid, freq=FALSE,
   main="Distribution of Studentized Residuals")
```




```
xfit<-seq(min(sresid),max(sresid),length=40)
yfit<-dnorm(xfit)
lines(xfit, yfit)
```



#Non-constant Error Variance

Evaluate homoscedasticity

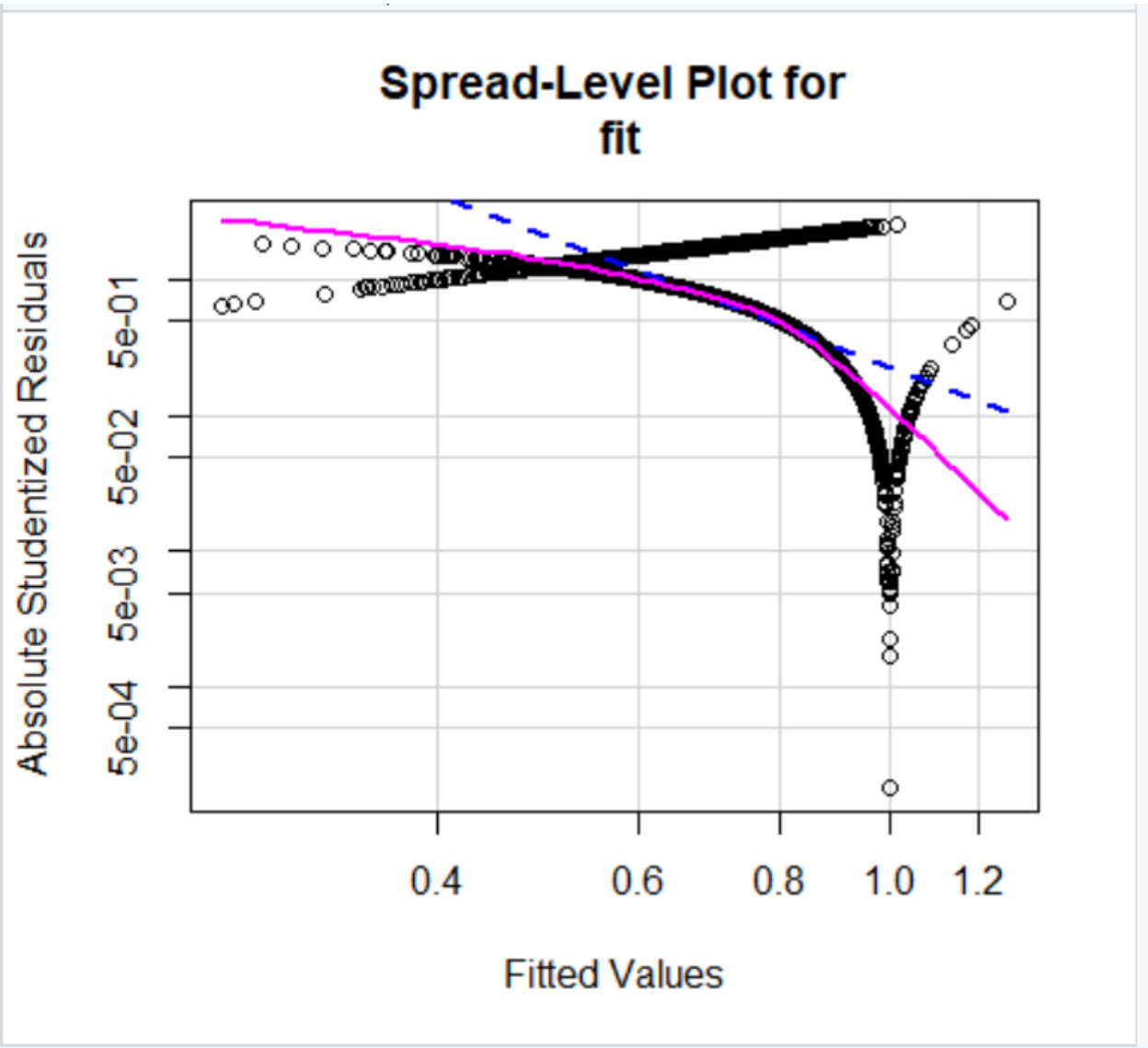
non-constant error variance test

ncvTest(fit)

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 295.9944, Df = 1, p = < 2.22e-16
```

plot studentized residuals vs. fitted values

spreadLevelPlot(fit)




```
Suggested power transformation:  4.179663
```

#Multi-collinearity

Evaluate Collinearity

vif(fit) # variance inflation factors

```
     loan_amnt    installment       int_rate        issue_d          grade        p
urpose            dti     emp_length home_ownership
     71.128436      64.966591      15.950003       1.072186      14.170535       1.
054222       1.159494       1.039202       1.084994
    annual_inc           term
      1.353647       7.050775
```

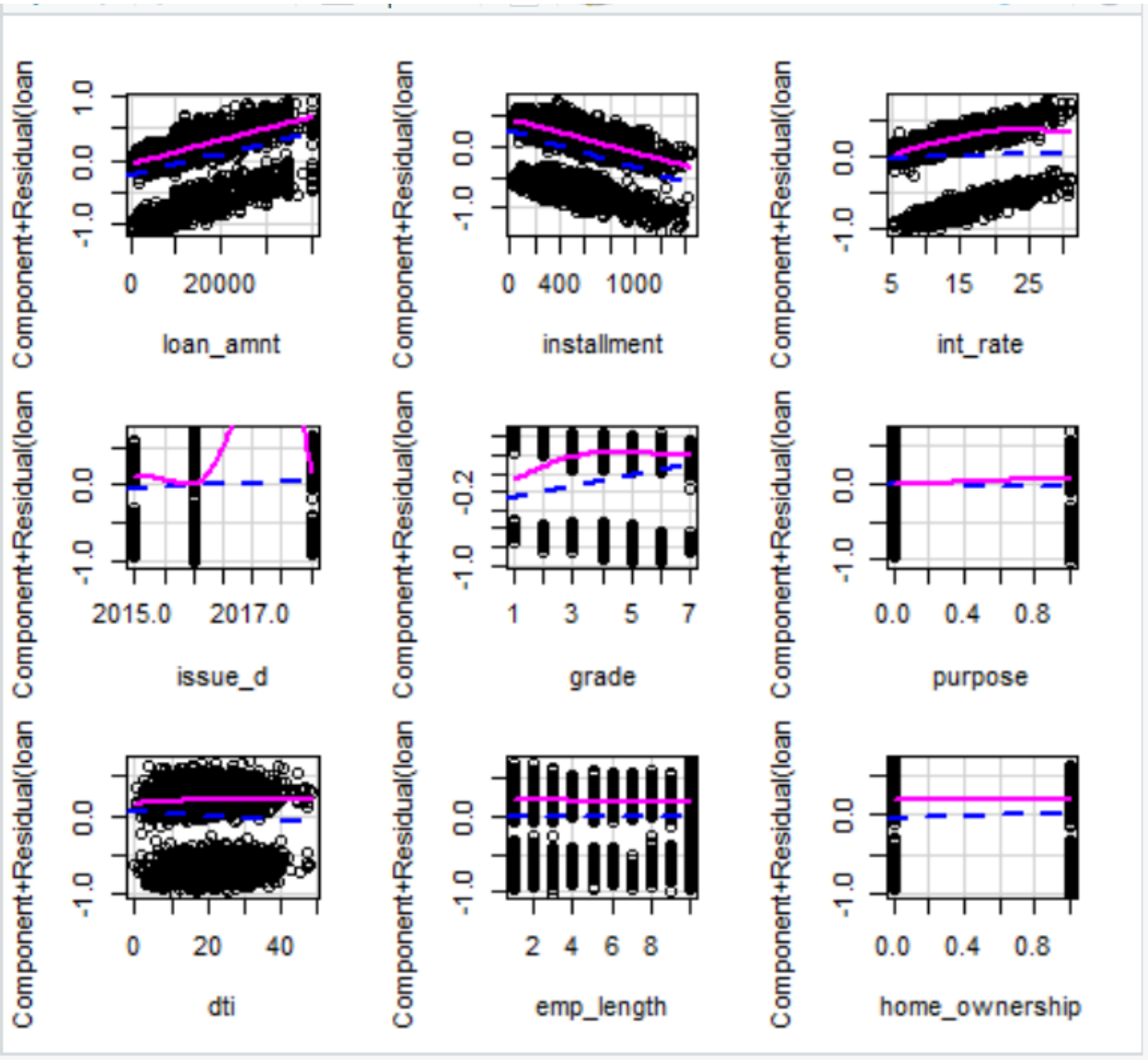
sqrt(vif(fit)) > 2 # problem?

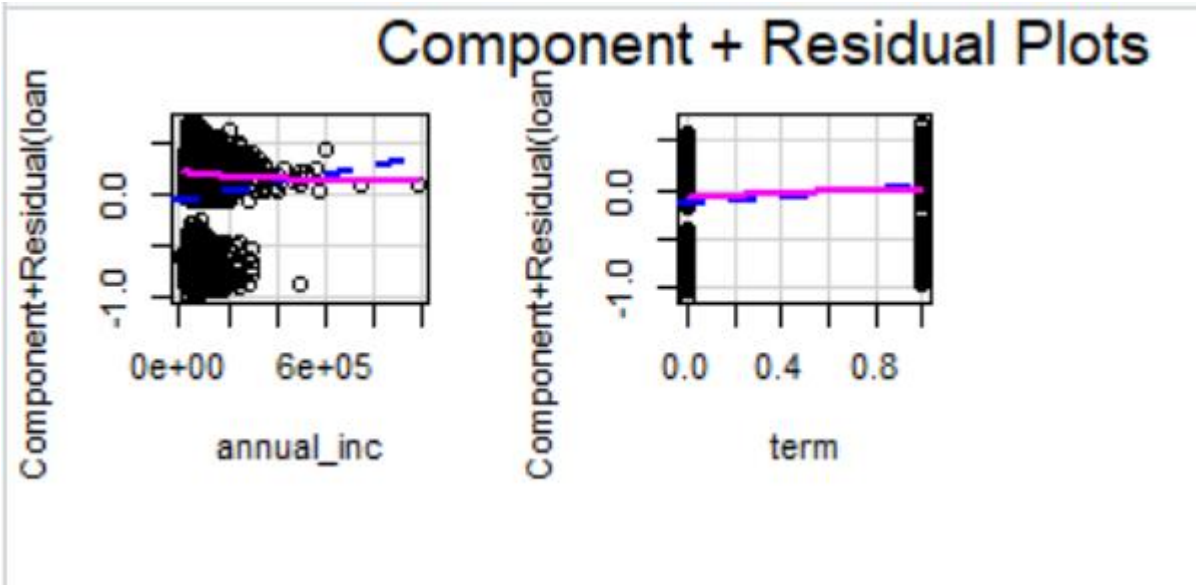
```
> sqrt(vif(fit)) > 2 # problem?
     loan_amnt    installment       int_rate        issue_d           grade         p
urpose             dti     emp_length home_ownership
          TRUE           TRUE           TRUE          FALSE            TRUE
 FALSE          FALSE          FALSE          FALSE
    annual_inc           term
         FALSE           TRUE
```

#Nonlinearity

component + residual plot

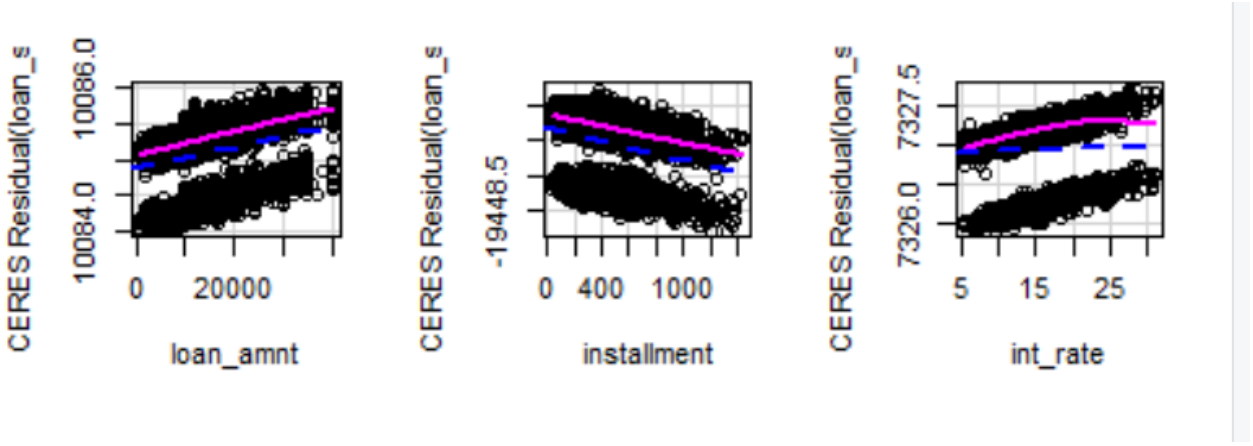
crPlots(fit)





Ceres plots

ceresPlots(fit)



#Non-independence of Errors

Test for Autocorrelated Errors

durbinWatsonTest(fit)

```
 lag Autocorrelation D-W Statistic p-value
   1     0.004416653       1.99021    0.74
 Alternative hypothesis: rho != 0
```

Global test of model assumptions

```
library(gvlma)

install.packages("gvlma", lib="/Library/Frameworks/R.framework/Versions/3.5/Resources/library")

library(gvlma)

gvmodel <- gvlma(fit)

summary(gvmodel)
```

```
Call:
lm(formula = loan_status ~ loan_amnt + installment + int_rate +
    issue_d + grade + purpose + dti + emp_length + home_ownership +
    annual_inc + term, data = mysample)

Residuals:
     Min       1Q   Median       3Q      Max
-1.01665 -0.00202  0.15740  0.25421  0.71990

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    -8.118e+01  1.690e+01  -4.802 1.62e-06 ***
loan_amnt       1.697e-05  5.430e-06   3.125 0.001789 **
installment    -6.011e-04  1.669e-04  -3.601 0.000320 ***
int_rate        2.922e-03  4.575e-03   0.639 0.523041
issue_d         4.040e-02  8.402e-03   4.808 1.57e-06 ***
grade           6.171e-02  1.731e-02   3.565 0.000367 ***
purpose        -2.987e-02  1.388e-02  -2.151 0.031521 *
dti            -2.779e-03  7.520e-04  -3.695 0.000222 ***
emp_length     -3.599e-04  1.600e-03  -0.225 0.822017
home_ownership  7.928e-02  1.236e-02   6.412 1.57e-10 ***
annual_inc      3.933e-07  1.271e-07   3.095 0.001978 **
term            1.757e-01  3.671e-02   4.786 1.75e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4036 on 4988 degrees of freedom
Multiple R-squared:  0.09166,   Adjusted R-squared:  0.08966
F-statistic: 45.76 on 11 and 4988 DF,  p-value: < 2.2e-16
```

```
ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
Level of Significance =  0.05

Call:
 gvlma(x = fit)

                     Value   p-value                   Decision
Global Stat        1.029e+03 0.000e+00 Assumptions NOT satisfied!
Skewness           1.009e+03 0.000e+00 Assumptions NOT satisfied!
Kurtosis           1.669e+01 4.408e-05 Assumptions NOT satisfied!
Link Function      3.183e+00 7.443e-02    Assumptions acceptable.
Heteroscedasticity 8.272e-02 7.736e-01    Assumptions acceptable.
```

fit

```
Call:
lm(formula = loan_status ~ loan_amnt + installment + int_rate +
    issue_d + grade + purpose + dti + emp_length + home_ownership +
    annual_inc + term, data = mysample)

Coefficients:
   (Intercept)       loan_amnt     installment        int_rate         issue_d
      grade         purpose             dti
    -8.118e+01       1.697e-05      -6.011e-04       2.922e-03       4.040e-02
  6.171e-02      -2.987e-02      -2.779e-03
    emp_length  home_ownership      annual_inc            term
    -3.599e-04       7.928e-02       3.933e-07       1.757e-01
```

summary(fit)

```
Call:
lm(formula = loan_status ~ loan_amnt + installment + int_rate +
    issue_d + grade + purpose + dti + emp_length + home_ownership +
    annual_inc + term, data = mysample)

Residuals:
     Min       1Q   Median       3Q      Max
-1.01665 -0.00202  0.15740  0.25421  0.71990

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    -8.118e+01  1.690e+01  -4.802 1.62e-06 ***
loan_amnt       1.697e-05  5.430e-06   3.125 0.001789 **
installment    -6.011e-04  1.669e-04  -3.601 0.000320 ***
int_rate        2.922e-03  4.575e-03   0.639 0.523041
issue_d         4.040e-02  8.402e-03   4.808 1.57e-06 ***
grade           6.171e-02  1.731e-02   3.565 0.000367 ***
purpose        -2.987e-02  1.388e-02  -2.151 0.031521 *
dti            -2.779e-03  7.520e-04  -3.695 0.000222 ***
emp_length     -3.599e-04  1.600e-03  -0.225 0.822017
home_ownership  7.928e-02  1.236e-02   6.412 1.57e-10 ***
annual_inc      3.933e-07  1.271e-07   3.095 0.001978 **
term            1.757e-01  3.671e-02   4.786 1.75e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4036 on 4988 degrees of freedom
Multiple R-squared:  0.09166,   Adjusted R-squared:  0.08966
F-statistic: 45.76 on 11 and 4988 DF,  p-value: < 2.2e-16
```

fit1 <- fit

fit2 <- lm(loan_status~loan_amnt+installment+

issue_d+grade+purpose+dti+emp_length+home_ownership+

annual_inc+term, data = mysample)

#Removing interest rate from the fit

compare models

Nested models can be compared using anova() function.

anova(fit1, fit2)

```
Analysis of Variance Table

Model 1: loan_status ~ loan_amnt + installment + int_rate + issue_d +
    grade + purpose + dti + emp_length + home_ownership + annual_inc +
    term
Model 2: loan_status ~ loan_amnt + installment + issue_d + grade + purpose +
    dti + emp_length + home_ownership + annual_inc + term
  Res.Df    RSS Df Sum of Sq     F Pr(>F)
1   4988 812.62
2   4989 812.69 -1 -0.066461 0.408  0.523
```

#add this library so your step AIC function will work

library(MASS)

Selecting a subset of predictor variables from a larger set (e.g., stepwise selection) is a controversial topic. You can perform stepwise selection (forward, backward, both) using the stepAIC( ) function from the MASS package. stepAIC( ) performs stepwise model selection by exact AIC.

step <- stepAIC(fit, direction="both")

```
Start:  AIC=-9060.65
loan_status ~ loan_amnt + installment + int_rate + issue_d +
    grade + purpose + dti + emp_length + home_ownership + annual_inc +
    term

                Df Sum of Sq    RSS     AIC
- emp_length     1    0.0082 812.63 -9062.6
- int_rate       1    0.0665 812.69 -9062.2
<none>                       812.62 -9060.7
- purpose        1    0.7538 813.37 -9058.0
- annual_inc     1    1.5607 814.18 -9053.1
- loan_amnt      1    1.5909 814.21 -9052.9
- grade          1    2.0711 814.69 -9049.9
- installment    1    2.1128 814.73 -9049.7
- dti            1    2.2242 814.84 -9049.0
- term           1    3.7320 816.35 -9039.7
- issue_d        1    3.7666 816.39 -9039.5
- home_ownership 1    6.6976 819.32 -9021.6

Step:  AIC=-9062.6
loan_status ~ loan_amnt + installment + int_rate + issue_d +
    grade + purpose + dti + home_ownership + annual_inc + term
```

```
                 Df Sum of Sq    RSS     AIC
- int_rate        1    0.0668 812.69 -9064.2
<none>                        812.63 -9062.6
+ emp_length      1    0.0082 812.62 -9060.7
- purpose         1    0.7584 813.39 -9059.9
- annual_inc      1    1.5584 814.19 -9055.0
- loan_amnt       1    1.5928 814.22 -9054.8
- grade           1    2.0721 814.70 -9051.9
- installment     1    2.1156 814.74 -9051.6
- dti             1    2.2405 814.87 -9050.8
- term            1    3.7355 816.36 -9041.7
- issue_d         1    3.7591 816.39 -9041.5
- home_ownership  1    6.8255 819.45 -9022.8

Step:  AIC=-9064.19
loan_status ~ loan_amnt + installment + issue_d + grade + purpose +
    dti + home_ownership + annual_inc + term

                 Df Sum of Sq    RSS     AIC
<none>                        812.69 -9064.2
+ int_rate        1    0.0668 812.63 -9062.6
+ emp_length      1    0.0086 812.69 -9062.2
- purpose         1    0.7480 813.44 -9061.6
- loan_amnt       1    1.5474 814.24 -9056.7
- annual_inc      1    1.5498 814.24 -9056.7
- installment     1    2.0975 814.79 -9053.3
- dti             1    2.2280 814.92 -9052.5
- term            1    3.8302 816.52 -9042.7
```

```
- issue_d         1    4.2536 816.95 -9040.1
- home_ownership  1    6.7998 819.49 -9024.5
- grade           1    8.5418 821.24 -9013.9
```

step$anova # display results

```
Stepwise Model Path
Analysis of Deviance Table

Initial Model:
loan_status ~ loan_amnt + installment + int_rate + issue_d +
    grade + purpose + dti + emp_length + home_ownership + annual_inc +
    term

Final Model:
loan_status ~ loan_amnt + installment + issue_d + grade + purpose +
    dti + home_ownership + annual_inc + term


         Step Df   Deviance Resid. Df Resid. Dev       AIC
1                                4988   812.6190 -9060.654
2 - emp_length  1 0.00824487      4989   812.6273 -9062.603
3   - int_rate  1 0.06676792      4990   812.6940 -9064.192
```

step$anova # display results

We can perform all-subsets regression using the leaps( ) function from the leaps package.

install.packages("leaps", lib="/Library/Frameworks/R.framework/Versions/3.5/Resources/library")

library(leaps)

leaps<-regsubsets(loan_status~loan_amnt+installment+

issue_d+grade+purpose+dti+emp_length+home_ownership+

annual_inc+term, data = mysample,nbest=10)

view results

summary(leaps) #tells us about the outliers

```
> summary(leaps) #tells us about the outliers
Subset selection object
Call: regsubsets.formula(loan_status ~ loan_amnt + installment + issue_d +
    grade + purpose + dti + emp_length + home_ownership + annual_inc +
    term, data = mysample, nbest = 10)
10 Variables  (and intercept)
              Forced in Forced out
loan_amnt         FALSE      FALSE
installment       FALSE      FALSE
issue_d           FALSE      FALSE
grade             FALSE      FALSE
purpose           FALSE      FALSE
dti               FALSE      FALSE
emp_length        FALSE      FALSE
home_ownership    FALSE      FALSE
annual_inc        FALSE      FALSE
term              FALSE      FALSE
10 subsets of each size up to 8
Selection Algorithm: exhaustive
          loan_amnt installment issue_d grade purpose dti emp_length home_ownership
 annual_inc term
1  ( 1 )  " "       " "          " "     "*"    " "     " " " " "         " "
 " "         " "
1  ( 2 )  " "       " "          " "     " "    " "     " " " " "         " "
 " "         "*"
1  ( 3 )  " "       " "          " "     " "    " "     "*" " " "         " "
 " "         " "
1  ( 4 )  " "       " "          " "     " "    " "     " " " " "         "*"
 " "         " "
1  ( 5 )  " "       " "          " "     " "    " "     " " " " "         " "
 "*"         " "
1  ( 6 )  " "       " "          "*"     " "    " "     " " " " "         " "
 " "         " "
1  ( 7 )  " "       "*"          " "     " "    " "     " " " " "         " "
 " "         " "
1  ( 8 )  "*"       " "          " "     " "    " "     " " " " "         " "
 " "         " "
1  ( 9 )  " "       " "          " "     " "    "*"     " " " " "         " "
 " "         " "
1  ( 10 ) " "       " "          " "     " "    " "     " " "*"  "         " "
 " "         " "
2  ( 1 )  " "       " "          " "     "*"    " "     " " " " "         "*"
 " "         " "
2  ( 2 )  " "       " "          "*"     "*"    " "     " " " " "         " "
 " "         " "
2  ( 3 )  " "       " "          " "     "*"    " "     "*" " " "         " "
 " "         " "
2  ( 4 )  " "       " "          " "     "*"    " "     " " " " "         " "
 "*"         " "
2  ( 5 )  " "       " "          " "     "*"    " "     " " " " "         " "
 " "         "*"
2  ( 6 )  "*"       " "          " "     "*"    " "     " " " " "         " "
 " "         " "
2  ( 7 )  " "       "*"          " "     "*"    " "     " " " " "         " "
 " "         " "
2  ( 8 )  " "       " "          " "     "*"    " "     " " "*"  "         " "
 " "         " "
```

```
2  ( 9 )  " "       " "       " "      "*"   "*"      " " " " " "       " "
 " "          " "
2  ( 10 ) " "       " "       " "      " "   " "      "*" " " " "       " "
 " "          "*"
3  ( 1 )  " "       " "       "*"      "*"   " "      " " " " " "       "*"
 " "          " "
3  ( 2 )  " "       " "       " "      "*"   " "      "*" " " " "       "*"
 " "          " "
3  ( 3 )  " "       " "       " "      "*"   " "      " " " " " "       "*"
 " "          "*"
3  ( 4 )  "*"       " "       " "      "*"   " "      " " " " " "       "*"
 " "          " "
3  ( 5 )  " "       " "       " "      "*"   " "      " " " " " "       "*"
 "*"          " "
3  ( 6 )  " "       " "       "*"      "*"   " "      "*" " " " "       " "
 " "          " "
3  ( 7 )  " "       "*"       " "      "*"   " "      " " " " " "       "*"
 " "          " "
3  ( 8 )  " "       " "       "*"      "*"   " "      " " " " " "       " "
 " "          "*"
3  ( 9 )  " "       " "       "*"      "*"   " "      " " " " " "       " "
 "*"          " "
3  ( 10 ) " "       " "       " "      "*"   "*"      " " " " " "       "*"
 " "          " "
4  ( 1 )  " "       " "       "*"      "*"   " "      " " " " " "       "*"
 " "          "*"
4  ( 2 )  " "       " "       "*"      "*"   " "      "*" " " " "       "*"
 " "          " "
4  ( 3 )  " "       " "       " "      "*"   " "      "*" " " " "       "*"
```

```
4  ( 4 )  "*"       " "       "*"      "*"   " "      " " " " "*"
 " "          " "
4  ( 5 )  " "       " "       "*"      "*"   " "      " " " " "*"
 "*"          " "
4  ( 6 )  " "       "*"       "*"      "*"   " "      " " " " "*"
 " "          " "
4  ( 7 )  " "       " "       "*"      "*"   "*"      " " " " "*"
 " "          " "
4  ( 8 )  " "       " "       "*"      "*"   " "      " " "*" "*"
 " "          " "
4  ( 9 )  "*"       " "       " "      "*"   " "      "*" " " "*"
 " "          " "
4  ( 10 ) "*"       " "       " "      "*"   " "      " " " " "*"
 "*"          " "
5  ( 1 )  " "       " "       "*"      "*"   " "      "*" " " "*"
 " "          "*"
5  ( 2 )  " "       " "       "*"      "*"   " "      " " " " "*"
 "*"          "*"
5  ( 3 )  "*"       " "       "*"      "*"   " "      "*" " " "*"
 " "          " "
5  ( 4 )  "*"       " "       "*"      "*"   " "      " " " " "*"
 "*"          " "
5  ( 5 )  " "       "*"       "*"      "*"   " "      "*" " " "*"
 " "          " "
5  ( 6 )  " "       "*"       "*"      "*"   " "      " " " " "*"
 " "          "*"
5  ( 7 )  " "       "*"       "*"      "*"   " "      " " " " "*"
 "*"          " "
5  ( 8 )  "*"       " "       "*"      "*"   " "      " " " " "*"
 " "          "*"
```

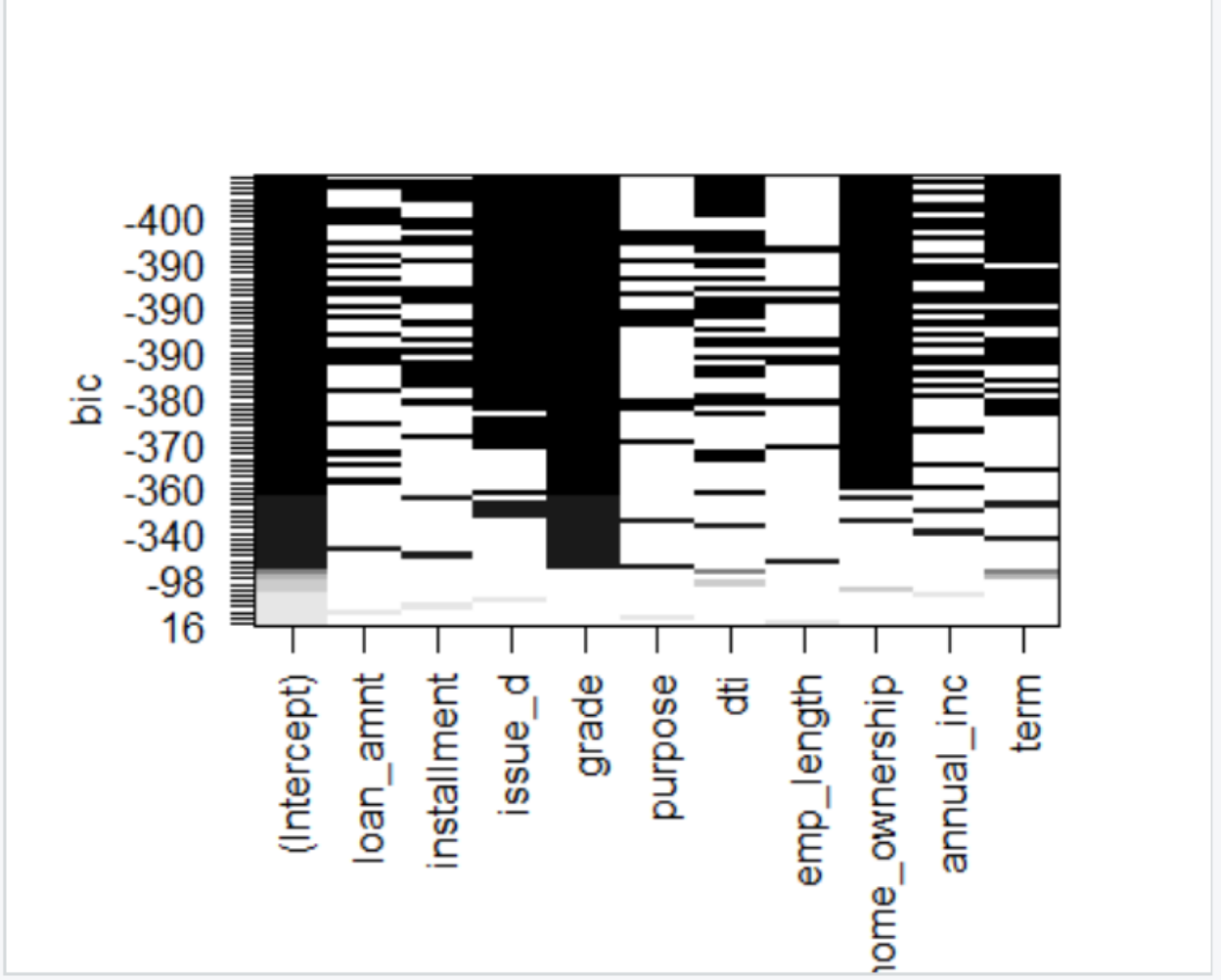
```
 " "          "*"
5  ( 9 )  " "        " "        "*"      "*"    " "      "*" " "        "*"
 "*"          " "
5  ( 10 ) " "        " "        "*"      "*"    "*"      " " " "        "*"
 " "          "*"
6  ( 1 )  " "        "*"        "*"      "*"    " "      "*" " "        "*"
 " "          "*"
6  ( 2 )  " "        " "        "*"      "*"    " "      "*" " "        "*"
 "*"          "*"
6  ( 3 )  "*"        " "        "*"      "*"    " "      "*" " "        "*"
 " "          "*"
6  ( 4 )  " "        "*"        "*"      "*"    " "      " " " "        "*"
 "*"          "*"
6  ( 5 )  " "        " "        "*"      "*"    "*"      "*" " "        "*"
 " "          "*"
6  ( 6 )  " "        " "        "*"      "*"    " "      "*" "*"        "*"
 " "          "*"
6  ( 7 )  "*"        " "        "*"      "*"    " "      " " " "        "*"
 "*"          "*"
6  ( 8 )  "*"        " "        "*"      "*"    " "      "*" " "        "*"
 "*"          " "
6  ( 9 )  "*"        "*"        "*"      "*"    " "      " " " "        "*"
 " "          "*"
6  ( 10 ) " "        "*"        "*"      "*"    " "      "*" " "        "*"
 "*"          " "
7  ( 1 )  "*"        "*"        "*"      "*"    " "      "*" " "        "*"
 " "          "*"
7  ( 2 )  " "        "*"        "*"      "*"    " "      "*" " "        "*"
 "*"          "*"
7  ( 3 )  "*"        " "        "*"      "*"    " "      "*" " "        "*"
 "*"          "*"
```

```
7  ( 4 )  "*"  "*"  "*"  "*"  " "  " "  " "  " "  "*"  "*"  "*"
7  ( 5 )  " "  "*"  "*"  "*"  "*"  "*"  " "  " "  "*"  " "  "*"
7  ( 6 )  " "  " "  "*"  "*"  "*"  "*"  " "  " "  "*"  "*"  "*"
7  ( 7 )  "*"  " "  "*"  "*"  "*"  "*"  " "  " "  "*"  " "  "*"
7  ( 8 )  " "  "*"  "*"  "*"  "*"  " "  " "  " "  "*"  "*"  "*"
7  ( 9 )  " "  "*"  "*"  "*"  " "  "*"  "*"  "*"  "*"  " "  "*"
7  ( 10 ) " "  " "  "*"  "*"  " "  "*"  "*"  "*"  "*"  "*"  "*"
8  ( 1 )  "*"  "*"  "*"  "*"  " "  "*"  " "  " "  "*"  "*"  "*"
8  ( 2 )  " "  "*"  "*"  "*"  "*"  "*"  " "  " "  "*"  "*"  "*"
8  ( 3 )  "*"  "*"  "*"  "*"  "*"  "*"  " "  " "  "*"  " "  "*"
8  ( 4 )  "*"  " "  "*"  "*"  "*"  "*"  " "  " "  "*"  "*"  "*"
8  ( 5 )  "*"  "*"  "*"  "*"  " "  "*"  "*"  "*"  "*"  " "  "*"
8  ( 6 )  "*"  "*"  "*"  "*"  "*"  " "  " "  " "  "*"  "*"  "*"
8  ( 7 )  " "  "*"  "*"  "*"  " "  "*"  "*"  "*"  "*"  "*"  "*"
8  ( 8 )  "*"  " "  "*"  "*"  " "  "*"  "*"  "*"  "*"  "*"  "*"
8  ( 9 )  "*"  "*"  "*"  "*"  " "  " "  "*"  "*"  "*"  "*"  "*"
8  ( 10 ) " "  "*"  "*"  "*"  "*"  "*"  "*"  "*"  "*"  " "  "*"
```

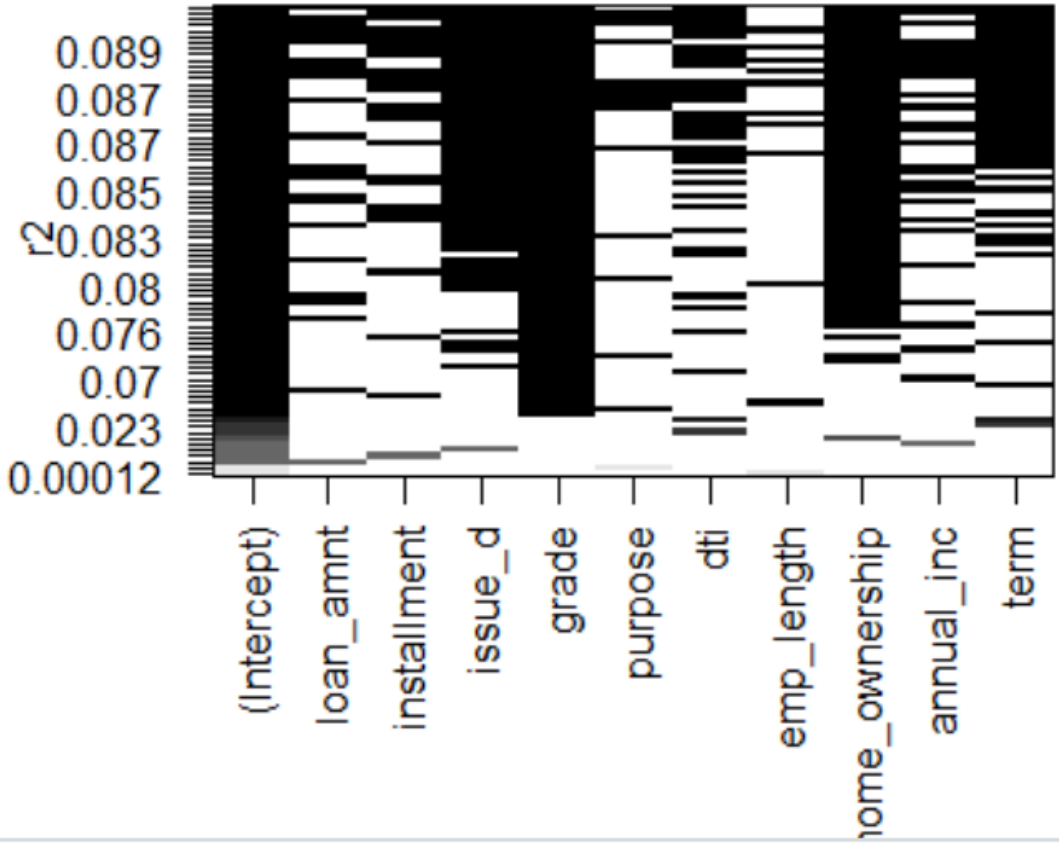
plot a table of models showing variables in each model.

models are ordered by the selection statistic.

plot(leaps)

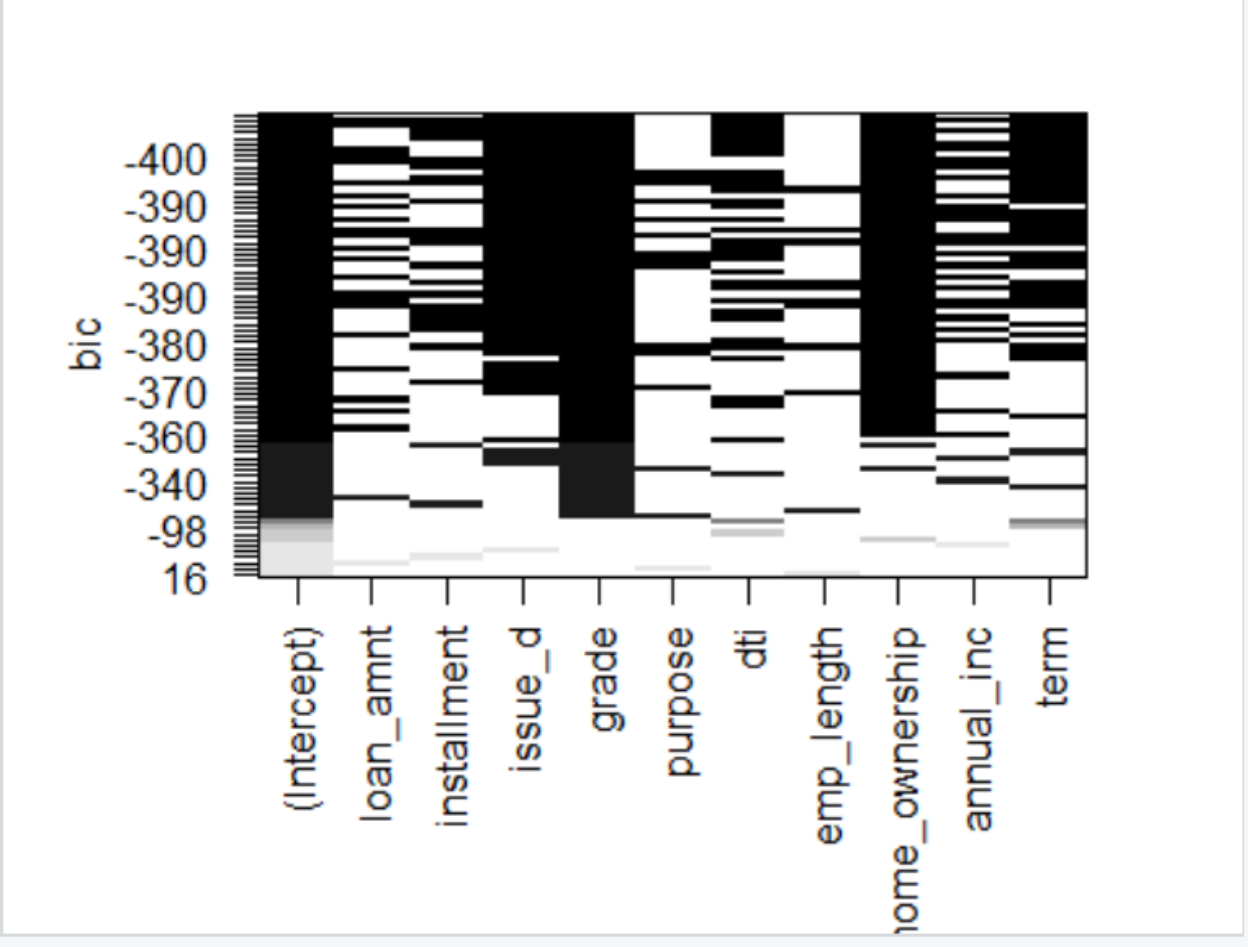



```
plot(leaps,scale="r2")
```



All Subsets Regression

plot(leaps,scale="bic")



summary(leaps)

leaps

```
> leaps
Subset selection object
Call: regsubsets.formula(loan_status ~ loan_amnt + installment + issue_d +
    grade + purpose + dti + emp_length + home_ownership + annual_inc +
    term, data = mysample, nbest = 10)
10 Variables  (and intercept)
              Forced in Forced out
loan_amnt         FALSE      FALSE
installment       FALSE      FALSE
issue_d           FALSE      FALSE
grade             FALSE      FALSE
purpose           FALSE      FALSE
dti               FALSE      FALSE
emp_length        FALSE      FALSE
home_ownership    FALSE      FALSE
annual_inc        FALSE      FALSE
term              FALSE      FALSE
10 subsets of each size up to 8
Selection Algorithm: exhaustive
```

coef(leaps,1:5)

```
[[1]]
(Intercept)       grade
 0.30817484  0.08869769

[[2]]
(Intercept)        term
  0.6458716   0.1543842

[[3]]
 (Intercept)          dti
 0.882623346 -0.006049171

[[4]]
   (Intercept) home_ownership
    0.70964187     0.08941621

[[5]]
 (Intercept)   annual_inc
7.147133e-01 6.231384e-07
```

```
# Calculate Relative Importance for Each Predictor
install.packages("relaimpo", lib="/Library/Frameworks/R.framework/Versions/3.5/Resources/library")
library(relaimpo)
calc.relimp(fit,type=c("lmg","last","first","pratt"),
        rela=TRUE)
```

```
Response variable: loan_status
Total response variance: 0.1789602
Analysis based on 5000 observations

11 Regressors:
loan_amnt installment int_rate issue_d grade purpose dti emp_length home_ownership
 annual_inc term
Proportion of variance explained by model: 9.17%
Metrics are normalized to sum to 100% (rela=TRUE).

Relative importance metrics:

                       lmg         last        first        pratt
loan_amnt      0.0358999865 0.0647113559 0.0257614282 -0.278038803
installment    0.0387679590 0.0859410732 0.0257788248  0.306327190
int_rate       0.2554295293 0.0027034069 0.3136494568 -0.093900289
issue_d        0.0584815604 0.1532106707 0.0292253559  0.055945873
grade          0.2941234605 0.0842436217 0.3399726874  0.514389643
purpose        0.0058943638 0.0306619850 0.0009454901  0.004463784
dti            0.0646984127 0.0904710073 0.0685418207  0.068466049
emp_length     0.0009153061 0.0003353719 0.0006194190 -0.000375158
home_ownership 0.0924196464 0.2724364032 0.0518354998  0.099946718
annual_inc     0.0422004566 0.0634817088 0.0297453879  0.040822109
term           0.1111693187 0.1518033955 0.1139246293  0.281952884
```

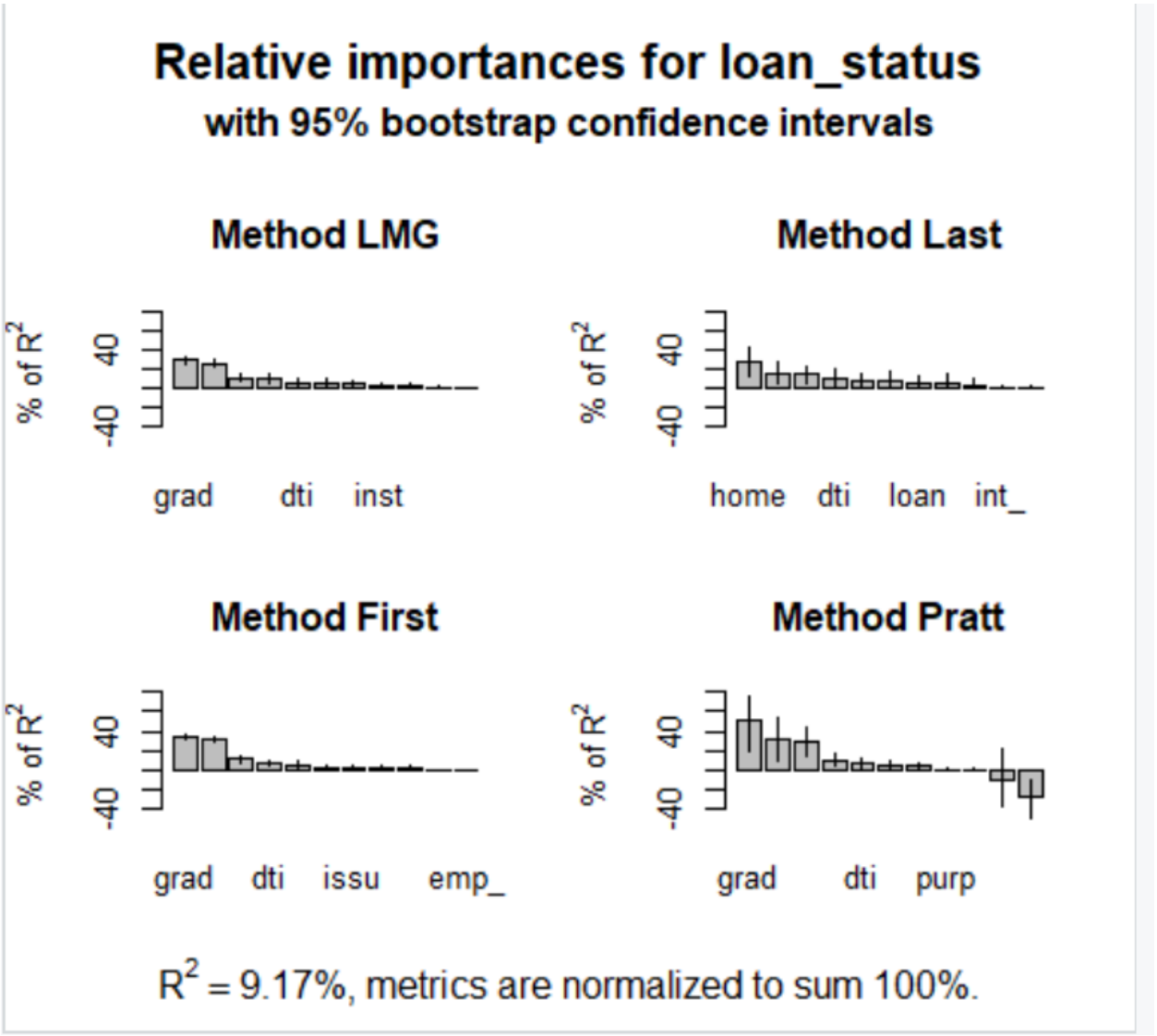
```
Average coefficients for different model sizes:

                          1X           2Xs           3Xs           4Xs           5X
s           6Xs           7Xs           8Xs
loan_amnt      -3.418808e-06 -2.912342e-06 -1.628562e-06 -1.200839e-08  1.690072e-0
6  3.393281e-06  5.146501e-06  7.106289e-06
installment    -1.100054e-04 -9.563677e-05 -1.097381e-04 -1.384510e-04 -1.739809e-0
4 -2.134340e-04 -2.578404e-04 -3.113341e-04
int_rate       -2.122727e-02 -1.860833e-02 -1.612968e-02 -1.376694e-02 -1.148971e-0
2 -9.261403e-03 -7.038997e-03 -4.772372e-03
issue_d         4.589438e-02  4.575060e-02  4.564335e-02  4.553999e-02  4.540143e-0
2  4.517620e-02  4.480163e-02  4.420818e-02
grade           8.869769e-02  8.691887e-02  8.470841e-02  8.204154e-02  7.894593e-0
2  7.550886e-02  7.188627e-02  6.831356e-02
purpose        -1.375718e-02 -1.664100e-02 -1.864848e-02 -2.000892e-02 -2.098438e-0
2 -2.184897e-02 -2.284971e-02 -2.416232e-02
dti            -6.049171e-03 -5.382494e-03 -4.781706e-03 -4.252218e-03 -3.801924e-0
3 -3.437574e-03 -3.162098e-03 -2.972678e-03
emp_length      1.292292e-03  1.162909e-03  1.003635e-03  8.296494e-04  6.511555e-0
4  4.735370e-04  2.988647e-04  1.275403e-04
home_ownership  8.941621e-02  8.791697e-02  8.602685e-02  8.401191e-02  8.212717e-0
2  8.057479e-02  7.948330e-02  7.889898e-02
annual_inc      6.231384e-07  6.284754e-07  6.134081e-07  5.852387e-07  5.510985e-0
7  5.168277e-07  4.862119e-07  4.605937e-07
term            1.543842e-01  1.336548e-01  1.210771e-01  1.141676e-01  1.113763e-0
1  1.119260e-01  1.156742e-01  1.229932e-01
```

```
                          9Xs           10Xs           11Xs
loan_amnt        9.514870e-06   1.268158e-05   1.696756e-05
installment     -3.804811e-04  -4.737475e-04  -6.010989e-04
int_rate        -2.403402e-03   1.353445e-04   2.922307e-03
issue_d          4.332413e-02   4.207841e-02   4.039961e-02
grade            6.511791e-02   6.273240e-02   6.171204e-02
purpose         -2.585064e-02  -2.783433e-02  -2.986655e-02
dti             -2.859335e-03  -2.803821e-03  -2.778593e-03
emp_length      -4.036844e-05  -2.039581e-04  -3.599147e-04
home_ownership   7.878096e-02   7.899256e-02   7.928059e-02
annual_inc       4.388727e-07   4.178996e-07   3.932509e-07
term             1.346653e-01   1.517897e-01   1.756997e-01
Warning message:
```

Bootstrap Measures of Relative Importance (1000 samples)

The relaimpo package gives the measures of relative importance for each of the predictors in the model. See help(calc.relimp) in the R console for details on the four measures of relative importance provided.

```
boot <- boot.relimp(fit, b = 1000, type = c("lmg",
                                "last", "first", "pratt"), rank = TRUE,
                diff = TRUE, rela = TRUE)
booteval.relimp(boot) # print result
plot(booteval.relimp(boot,sort=TRUE)) # plot result
```


Relative importances for loan_status
with 95% bootstrap confidence intervals
Method LMG
% of R2
40
-40
grad
dti
inst
Method Last
% of R2
40
-40
home
dti
loan
int_
Method First
% of R2
40
-40
grad
dti
issu
emp_
Method Pratt
% of R2
40
-40
grad
dti
purp
R2 = 9.17%, metrics are normalized to sum 100%.

summary(fit)

```
Call:
lm(formula = loan_status ~ loan_amnt + installment + int_rate +
    issue_d + grade + purpose + dti + emp_length + home_ownership +
    annual_inc + term, data = mysample)

Residuals:
     Min       1Q   Median       3Q      Max
-1.01665 -0.00202  0.15740  0.25421  0.71990

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)    -8.118e+01  1.690e+01  -4.802 1.62e-06 ***
loan_amnt       1.697e-05  5.430e-06   3.125 0.001789 **
installment    -6.011e-04  1.669e-04  -3.601 0.000320 ***
int_rate        2.922e-03  4.575e-03   0.639 0.523041
issue_d         4.040e-02  8.402e-03   4.808 1.57e-06 ***
grade           6.171e-02  1.731e-02   3.565 0.000367 ***
purpose        -2.987e-02  1.388e-02  -2.151 0.031521 *
dti            -2.779e-03  7.520e-04  -3.695 0.000222 ***
emp_length     -3.599e-04  1.600e-03  -0.225 0.822017
home_ownership  7.928e-02  1.236e-02   6.412 1.57e-10 ***
annual_inc      3.933e-07  1.271e-07   3.095 0.001978 **
term            1.757e-01  3.671e-02   4.786 1.75e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4036 on 4988 degrees of freedom
Multiple R-squared:  0.09166,   Adjusted R-squared:  0.08966
F-statistic: 45.76 on 11 and 4988 DF,  p-value: < 2.2e-16
```

From the above observations and value of R^2, we can clearly see that a regression prediction model won’t fit our dataset and hence we will proceed with other models.