Multiple Regression

10/28/2020

Loading required libraries

```
library(cluster)
library(data.table)
library(magrittr)
library(stringr)
library(ggplot2)
library(knitr)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------ tidyverse 1.3.0 --
```

```
## v tibble  3.0.3     v purrr   0.3.4
## v tidyr   1.1.2     v dplyr   1.0.2
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts --------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::set_names() masks magrittr::set_names()
## x purrr::transpose() masks data.table::transpose()
```

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(FactoMineR)
library(nFactors)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'nFactors'
```

```
## The following object is masked from 'package:lattice':
##
##     parallel
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(gvlma)
library(leaps)
library(relaimpo)
```

```
## Loading required package: boot
```

```
##
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:lattice':
##
##     melanoma
```

```
## The following object is masked from 'package:psych':
##
##     logit
```

```
## Loading required package: survey
```

```
## Loading required package: grid

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loading required package: survival

##
## Attaching package: 'survival'

## The following object is masked from 'package:boot':
##
##     aml

##
## Attaching package: 'survey'

## The following object is masked from 'package:graphics':
##
##     dotchart

## Loading required package: mitools

## This is the global version of package relaimpo.

## If you are a non-US user, a version with the interesting additional metric pmvd is available

## from Ulrike Groempings web site at prof.beuth-hochschule.de/groemping.
```

```
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:boot':
##
##     logit

## The following object is masked from 'package:psych':
##
##     logit
```

```
## The following object is masked from 'package:dplyr':
##
##     recode

## The following object is masked from 'package:purrr':
##
##     some
```

```
#library(FFally)
```

Data Loading

```
Lending_Data <- read_csv('Lending_Data.csv')
```

```
## Parsed with column specification:
## cols(
##   member_id = col_character(),
##   loan_status = col_character(),
##   int_rate = col_character(),
##   Bin_int = col_double(),
##   dti = col_double(),
##   Bin_dti = col_double(),
##   Default_flag = col_double(),
##   No_of_Enquiry = col_double(),
##   enq_buckets = col_character(),
##   annual_inc = col_double(),
##   Income_bins = col_double(),
##   home_ownership = col_character(),
##   purpose = col_character(),
##   open_acc = col_double(),
##   emp_length = col_character(),
##   verification_status = col_character(),
##   delinq_2yrs = col_double(),
##   loan_amnt = col_double(),
##   Bins_loan_amt = col_double()
## )
```

```
Lend = copy(Lending_Data)
Lend = setDT(Lend)
#view(Lend)
str(Lend)
```

```
## Classes 'data.table' and 'data.frame':   35808 obs. of  19 variables:
##  $ member_id          : chr  "LC1" "LC10" "LC100" "LC1000" ...
##  $ loan_status        : chr  "Charged Off" "Fully Paid" "Fully Paid" "Fully Paid" ...
##  $ int_rate           : chr  "11.71%" "15.96%" "10.65%" "12.69%" ...
##  $ Bin_int            : num  10 16 8 11 22 1 23 10 5 16 ...
##  $ dti                : num  1.06 2.61 11.34 14 13.01 ...
##  $ Bin_dti            : num  2 3 11 14 13 11 5 10 24 14 ...
##  $ Default_flag       : num  1 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ No_of_Enquiry      : num  0 1 1 1 0 0 3 0 1 2 ...
##  $ enq_buckets        : chr  "0" "1-4" "1-4" "1-4" ...
##  $ annual_inc         : num  110000 135000 75000 51000 41500 ...
##  $ Income_bins        : num  9 11 6 4 3 4 12 7 6 4 ...
##  $ home_ownership     : chr  "MORTGAGE" "RENT" "MORTGAGE" "RENT" ...
##  $ purpose            : chr  "credit_card" "other" "educational" "credit_card" ...
##  $ open_acc           : num  6 3 7 5 8 5 4 7 6 9 ...
##  $ emp_length         : chr  "LT 1year" "10+ years" "2 years" "1 year" ...
##  $ verification_status: chr  "Not Verified" "Source Verified" "Source Verified" "Source Verified" ..
##  $ delinq_2yrs        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ loan_amnt          : num  7000 2000 12000 9350 6000 ...
##  $ Bins_loan_amt      : num  6 2 10 8 5 8 5 10 2 8 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   member_id = col_character(),
##   ..   loan_status = col_character(),
##   ..   int_rate = col_character(),
##   ..   Bin_int = col_double(),
##   ..   dti = col_double(),
##   ..   Bin_dti = col_double(),
##   ..   Default_flag = col_double(),
##   ..   No_of_Enquiry = col_double(),
##   ..   enq_buckets = col_character(),
##   ..   annual_inc = col_double(),
##   ..   Income_bins = col_double(),
##   ..   home_ownership = col_character(),
##   ..   purpose = col_character(),
##   ..   open_acc = col_double(),
##   ..   emp_length = col_character(),
##   ..   verification_status = col_character(),
##   ..   delinq_2yrs = col_double(),
##   ..   loan_amnt = col_double(),
##   ..   Bins_loan_amt = col_double()
##   .. )
##  - attr(*, ".internal.selfref")=<externalptr>
```

Data Cleaning

```
Lend[, member_id := factor(member_id)]
Lend[, loan_status := factor(loan_status)]
Lend[, home_ownership := factor(home_ownership)]
Lend[, purpose := factor(purpose)]
Lend[, verification_status := factor(verification_status)]

Lend[, int_rate := gsub('[%]', '', int_rate)]
Lend[, int_rate := trimws(int_rate)]
Lend[, int_rate := suppressWarnings(as.numeric(int_rate))]

Lend[open_acc %in% c(1, 2, 3, 4, 5), 'x' := 'LT5']
Lend[open_acc %in% c(6, 7, 8, 9, 10), 'x' := '6-10']
Lend[open_acc %in% c(11, 12, 13, 14, 15), 'x' := '11-15']
Lend[open_acc > 15, 'x' := '15+']
```

```r
Lend = Lend %>% rename(no_of_acct = x)
str(Lend)
```

```
## Classes 'data.table' and 'data.frame':   35808 obs. of  20 variables:
##  $ member_id          : Factor w/ 35808 levels "LC1","LC10","LC100",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ loan_status        : Factor w/ 2 levels "Charged Off",..: 1 2 2 2 2 2 2 2 2 2 ...
##  $ int_rate           : num  11.7 16 10.7 12.7 19.7 ...
##  $ Bin_int            : num  10 16 8 11 22 1 23 10 5 16 ...
##  $ dti                : num  1.06 2.61 11.34 14 13.01 ...
##  $ Bin_dti            : num  2 3 11 14 13 11 5 10 24 14 ...
##  $ Default_flag       : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ No_of_Enquiry      : num  0 1 1 1 0 0 3 0 1 2 ...
##  $ enq_buckets        : chr  "0" "1-4" "1-4" "1-4" ...
##  $ annual_inc         : num  110000 135000 75000 51000 41500 ...
##  $ Income_bins        : num  9 11 6 4 3 4 12 7 6 4 ...
##  $ home_ownership     : Factor w/ 5 levels "MORTGAGE","NONE",..: 1 5 1 5 1 1 1 5 5 1 ...
##  $ purpose            : Factor w/ 14 levels "car","credit_card",..: 2 10 4 2 3 3 8 2 10 3 ...
##  $ open_acc           : num  6 3 7 5 8 5 4 7 6 9 ...
##  $ emp_length         : chr  "LT 1year" "10+ years" "2 years" "1 year" ...
##  $ verification_status: Factor w/ 3 levels "Not Verified",..: 1 2 2 2 3 3 1 1 1 2 ...
##  $ delinq_2yrs        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ loan_amnt          : num  7000 2000 12000 9350 6000 ...
##  $ Bins_loan_amt      : num  6 2 10 8 5 8 5 10 2 8 ...
##  $ no_of_acct         : chr  "6-10" "LT5" "6-10" "LT5" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   member_id = col_character(),
##   ..   loan_status = col_character(),
##   ..   int_rate = col_character(),
##   ..   Bin_int = col_double(),
##   ..   dti = col_double(),
##   ..   Bin_dti = col_double(),
##   ..   Default_flag = col_double(),
##   ..   No_of_Enquiry = col_double(),
##   ..   enq_buckets = col_character(),
##   ..   annual_inc = col_double(),
##   ..   Income_bins = col_double(),
##   ..   home_ownership = col_character(),
##   ..   purpose = col_character(),
##   ..   open_acc = col_double(),
##   ..   emp_length = col_character(),
##   ..   verification_status = col_character(),
##   ..   delinq_2yrs = col_double(),
##   ..   loan_amnt = col_double(),
##   ..   Bins_loan_amt = col_double()
##   .. )
##  - attr(*, ".internal.selfref")=<externalptr>
##  - attr(*, "index")= int
##   ..- attr(*, "__open_acc")= int  75 113 157 195 377 382 458 611 628 642 ...
```

Data Splitting

```
#Training Testing

## 10% of the sample size
smp_size = floor(0.10 * nrow(Lend))

## set the seed to make our partition reproducible
set.seed(123)
train_ind = sample(seq_len(nrow(Lend)), size = smp_size)

train = Lend[train_ind, ]
test = Lend[-train_ind, ]
```

Multiple Regression

Creating proper data for regression:

```
Lend_reg = train[, c(7, 3, 5, 8, 10, 14, 18)]
head(Lend_reg)
```

```
##    Default_flag int_rate   dti No_of_Enquiry annual_inc open_acc loan_amnt
## 1:            0    14.54 15.94             0      43200       16     19000
## 2:            0    17.99 19.04             0     105000        7      5000
## 3:            0     7.49 12.70             2     111000        7      5750
## 4:            0    10.00  8.88             1      40800       25      3900
## 5:            0     9.99 11.06             1      60000        8      6000
## 6:            0    13.49  6.32             1     120000        7      2500
```

Performing multiple regression on Lending dataset

```
fit <- lm(Default_flag ~ int_rate + dti + annual_inc + loan_amnt, data = Lend_reg)
```

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).

Showing the results:

```
summary(fit)
```

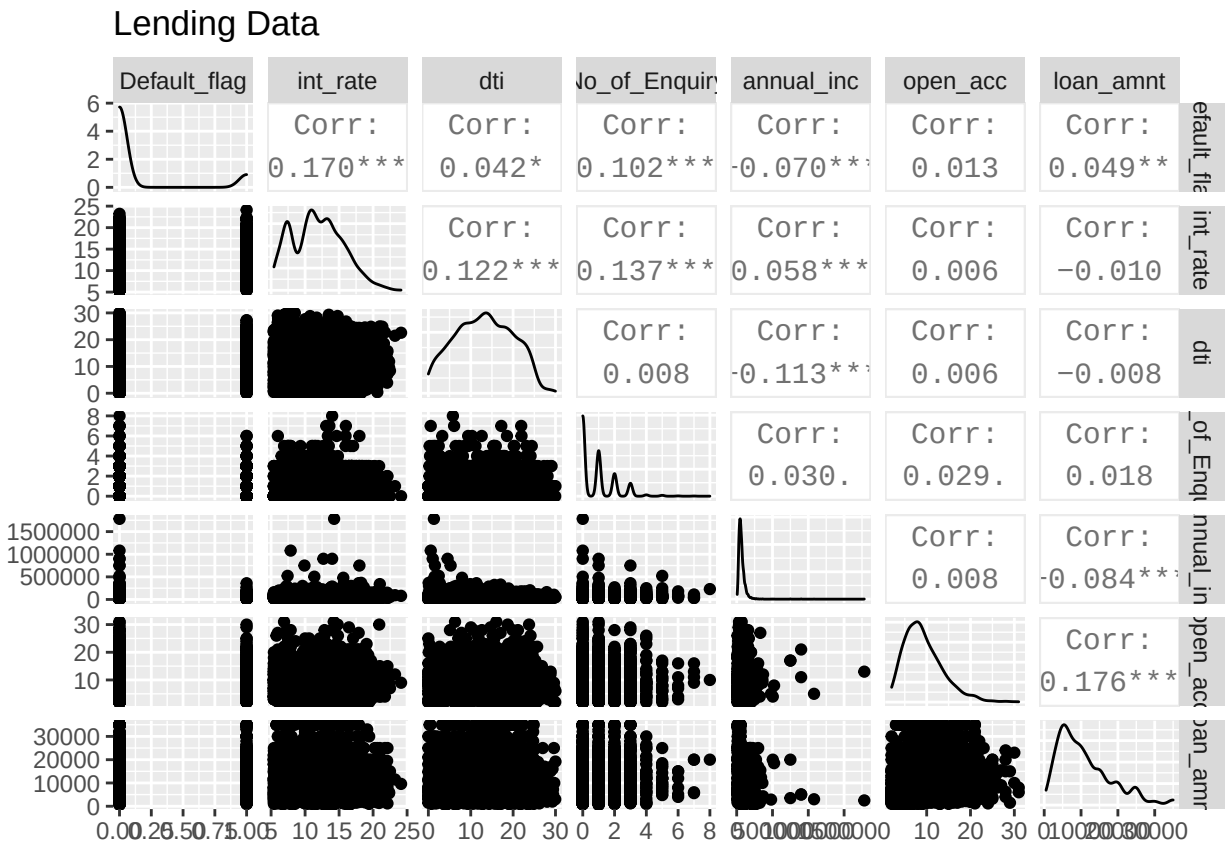
```
##
## Call:
## lm(formula = Default_flag ~ int_rate + dti + annual_inc + loan_amnt,
##     data = Lend_reg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32002 -0.16833 -0.11749 -0.05753  0.98888
##
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.611e-02  2.391e-02  -2.347  0.01900 *
## int_rate     1.618e-02  1.550e-03  10.442  < 2e-16 ***
## dti          6.462e-04  8.537e-04   0.757  0.44912
## annual_inc  -4.385e-07  9.670e-08  -4.535 5.95e-06 ***
## loan_amnt    2.103e-06  7.782e-07   2.702  0.00692 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3372 on 3575 degrees of freedom
## Multiple R-squared:  0.03742,    Adjusted R-squared:  0.03634
## F-statistic: 34.74 on 4 and 3575 DF,  p-value: < 2.2e-16
```

```
coefficients(fit)
```

```
##   (Intercept)      int_rate           dti    annual_inc     loan_amnt
## -5.610807e-02  1.618029e-02  6.462005e-04 -4.385255e-07  2.102746e-06
```

```
ggpairs(data = Lend_reg, title = "Lending Data")
```




```
confint(fit, level = 0.95)
```

```
##                      2.5 %        97.5 %
## (Intercept) -1.029877e-01 -9.228447e-03
```

```
## int_rate     1.314228e-02  1.921829e-02
## dti         -1.027549e-03  2.319950e-03
## annual_inc  -6.281086e-07 -2.489423e-07
## loan_amnt    5.769823e-07  3.628510e-06
```

Predicted Values

```
head(fitted(fit))
```

```
##          1          2          3          4          5          6
## 0.21046160 0.21174748 0.03670348 0.10174192 0.09898491 0.11888178
```

```
head(residuals(fit))
```

```
##           1           2           3           4           5           6
## -0.21046160 -0.21174748 -0.03670348 -0.10174192 -0.09898491 -0.11888178
```

Anova Table:

```
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: Default_flag
##              Df Sum Sq Mean Sq  F value    Pr(>F)
## int_rate      1  12.18 12.1752 107.1006 < 2.2e-16 ***
## dti           1   0.19  0.1925   1.6933  0.193257
## annual_inc    1   2.60  2.6009  22.8786 1.795e-06 ***
## loan_amnt     1   0.83  0.8300   7.3011  0.006924 **
## Residuals  3575 406.41  0.1137
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
vcov(fit)
```

```
##               (Intercept)      int_rate           dti    annual_inc
## (Intercept)  5.717127e-04 -2.544677e-05 -8.355625e-06 -7.199771e-10
## int_rate    -2.544677e-05  2.400968e-06 -1.716986e-07 -1.090139e-11
## dti         -8.355625e-06 -1.716986e-07  7.287703e-07  1.009430e-11
## annual_inc  -7.199771e-10 -1.090139e-11  1.009430e-11  9.349951e-15
## loan_amnt   -7.249051e-09  3.011267e-12  1.122032e-11  6.421269e-15
##                 loan_amnt
## (Intercept) -7.249051e-09
## int_rate     3.011267e-12
## dti          1.122032e-11
## annual_inc   6.421269e-15
## loan_amnt    6.055979e-13
```

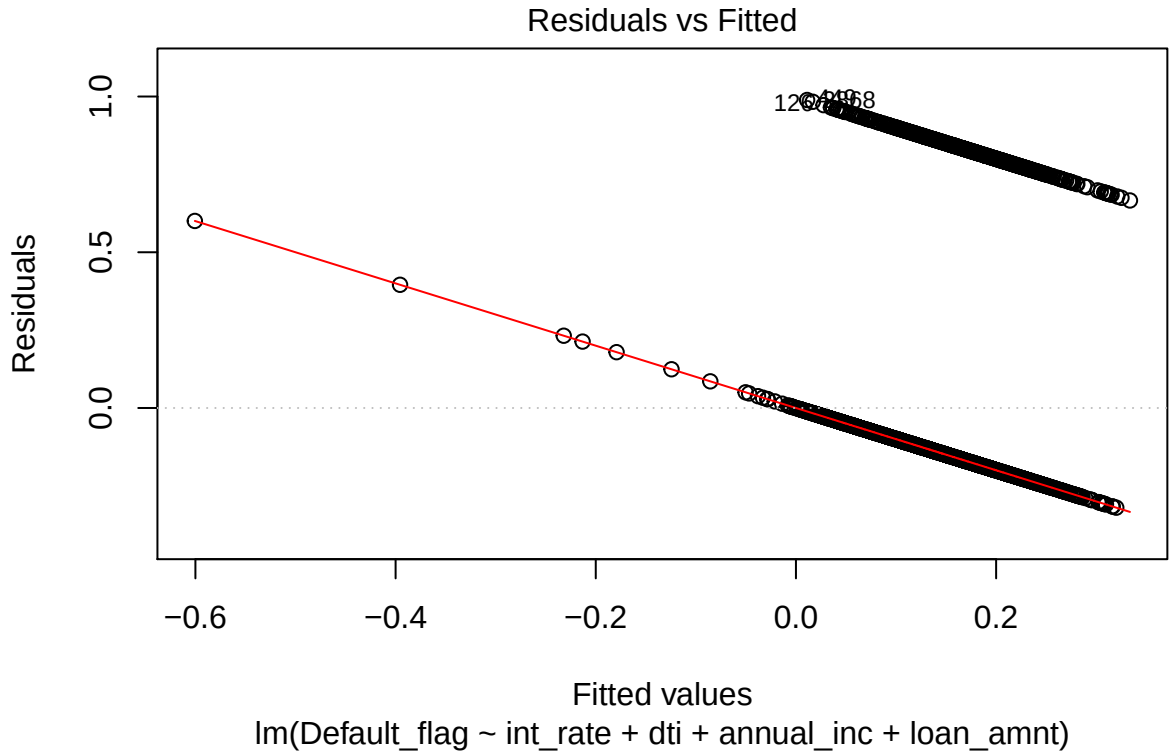
```
cov2cor(vcov(fit))
```

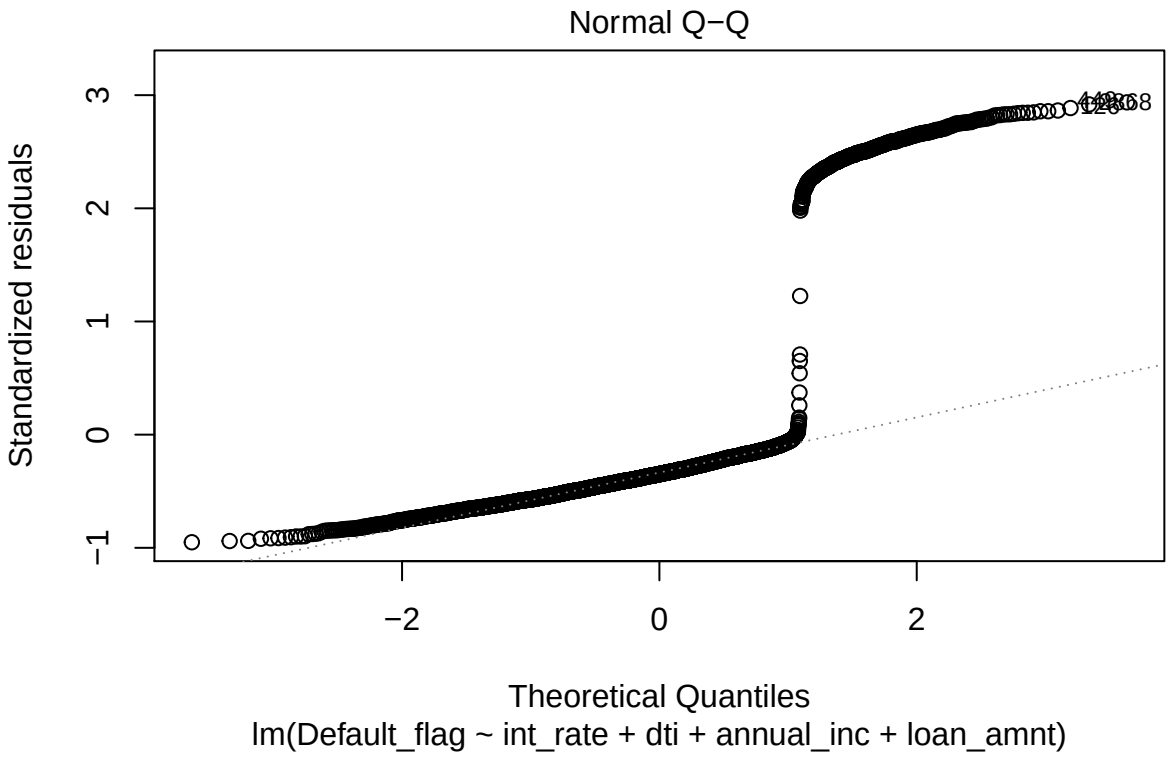
```
##             (Intercept)     int_rate         dti  annual_inc    loan_amnt
## (Intercept)   1.0000000 -0.686831946 -0.40934988 -0.31140465 -0.389583032
## int_rate     -0.6868319  1.000000000 -0.12980104 -0.07275856  0.002497261
## dti          -0.4093499 -0.129801042  1.00000000  0.12228580  0.016889535
## annual_inc   -0.3114047 -0.072758556  0.12228580  1.00000000  0.085334383
## loan_amnt    -0.3895830  0.002497261  0.01688953  0.08533438  1.000000000
```

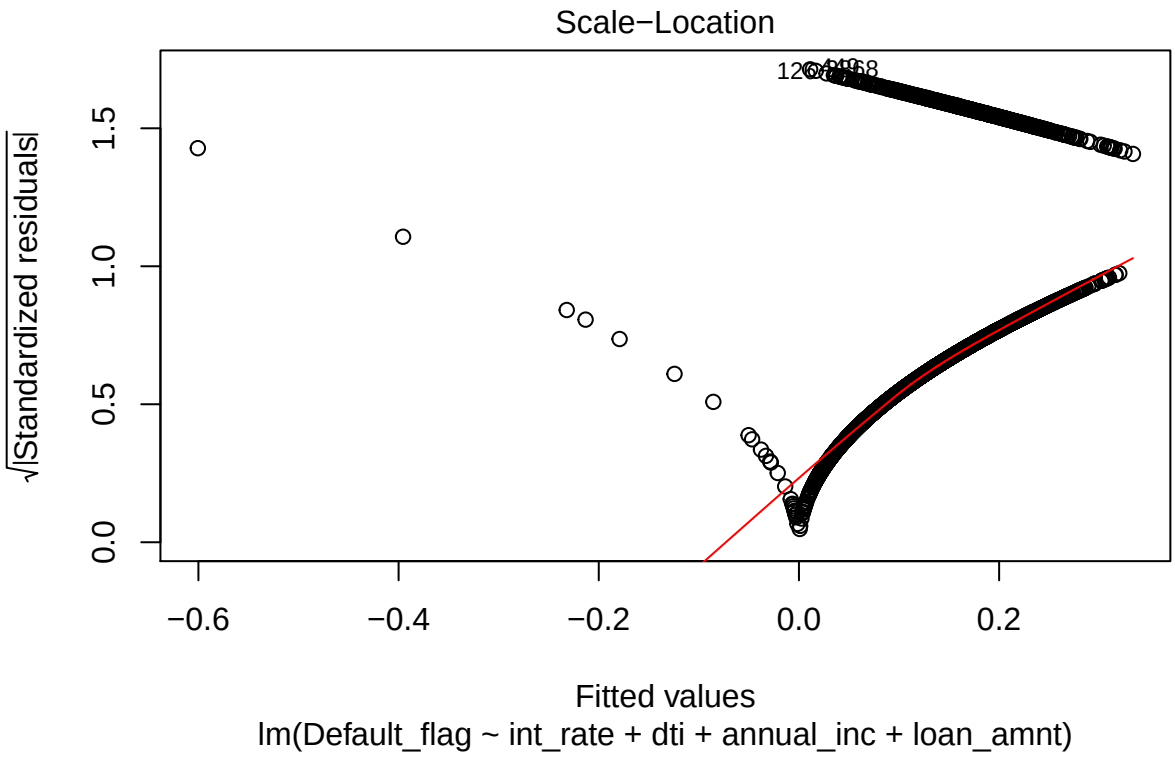
```
temp <- influence.measures(fit)
```

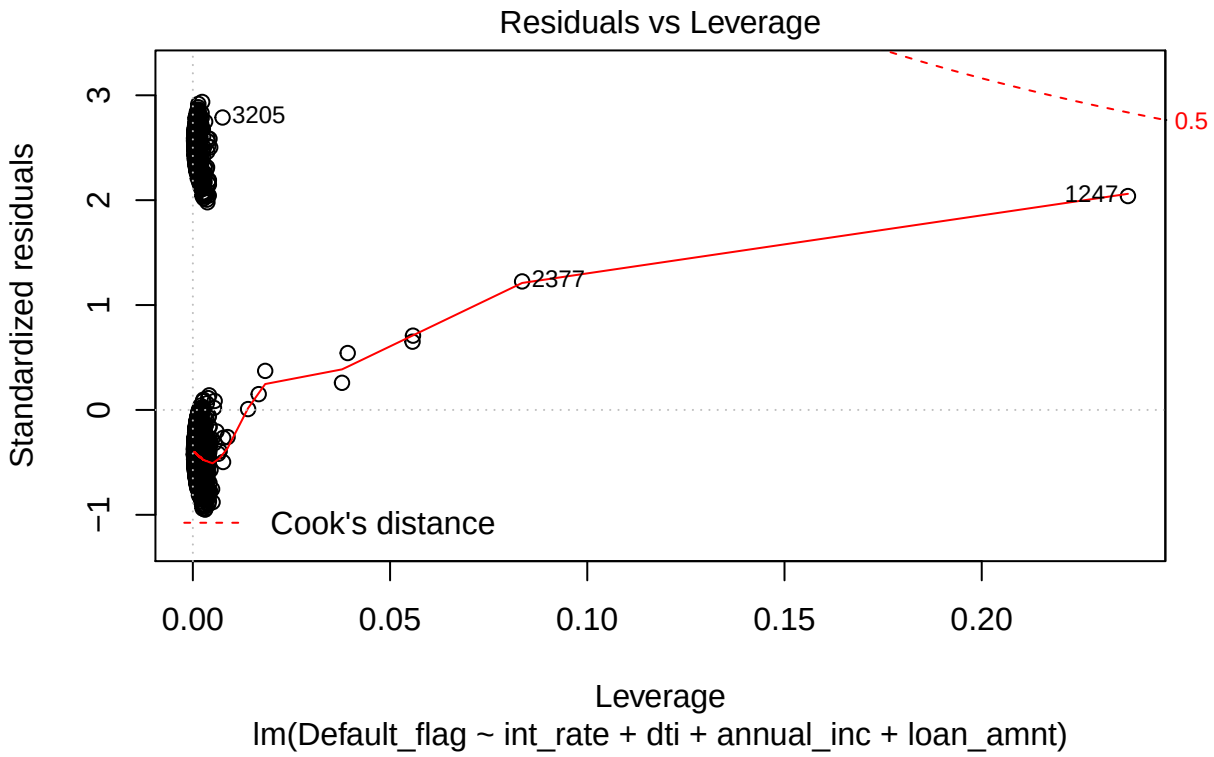
Diagnostic plots:

```
plot(fit)
```




Normal Q−Q
Standardized residuals
−1
0
1
2
3
−2
0
2
Theoretical Quantiles
lm(Default_flag ~ int_rate + dti + annual_inc + loan_amnt)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
−0.6
−0.4
−0.2
0.0
0.2
Fitted values
lm(Default_flag ~ int_rate + dti + annual_inc + loan_amnt)



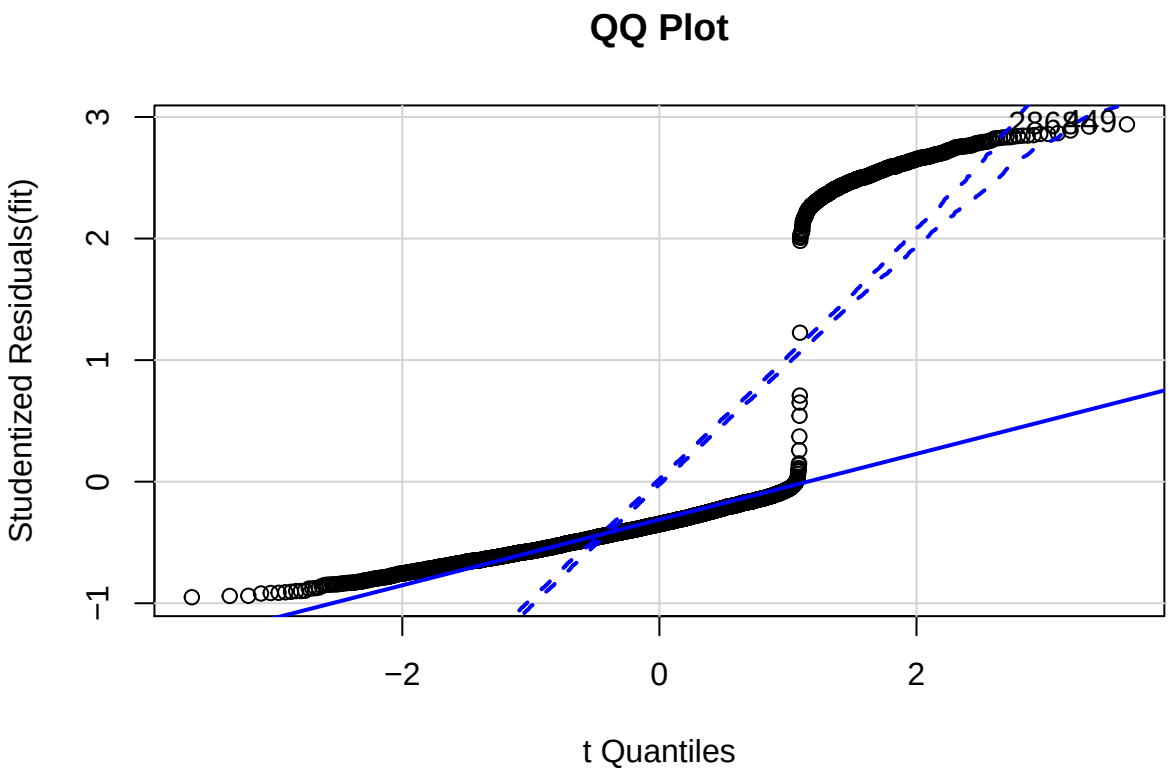
Assessing Outliers:

```
outlierTest(fit)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferroni p
## 449 2.939542           0.003308           NA
```

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

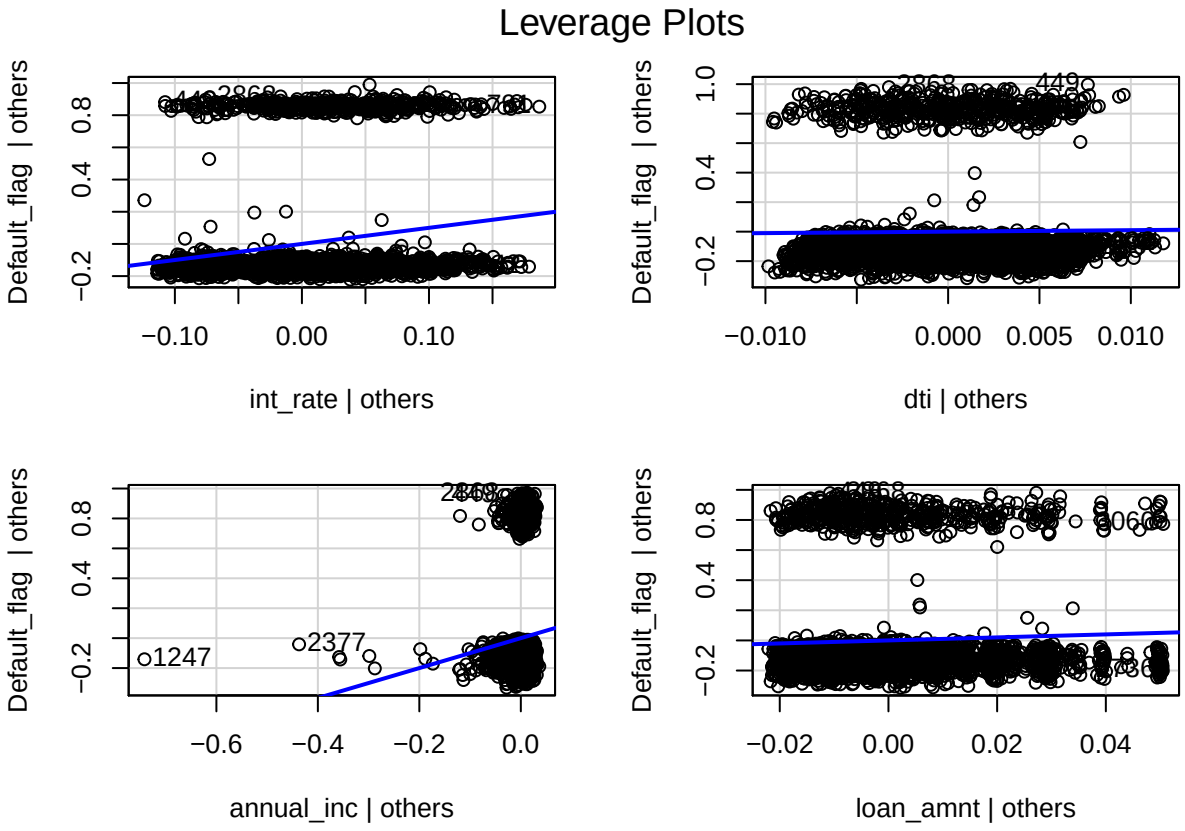
```
## Warning in rlm.default(x, y, weights, method = method, wt.method = wt.method, :
## 'rlm' failed to converge in 20 steps
```




```
## [1]  449 2868
```

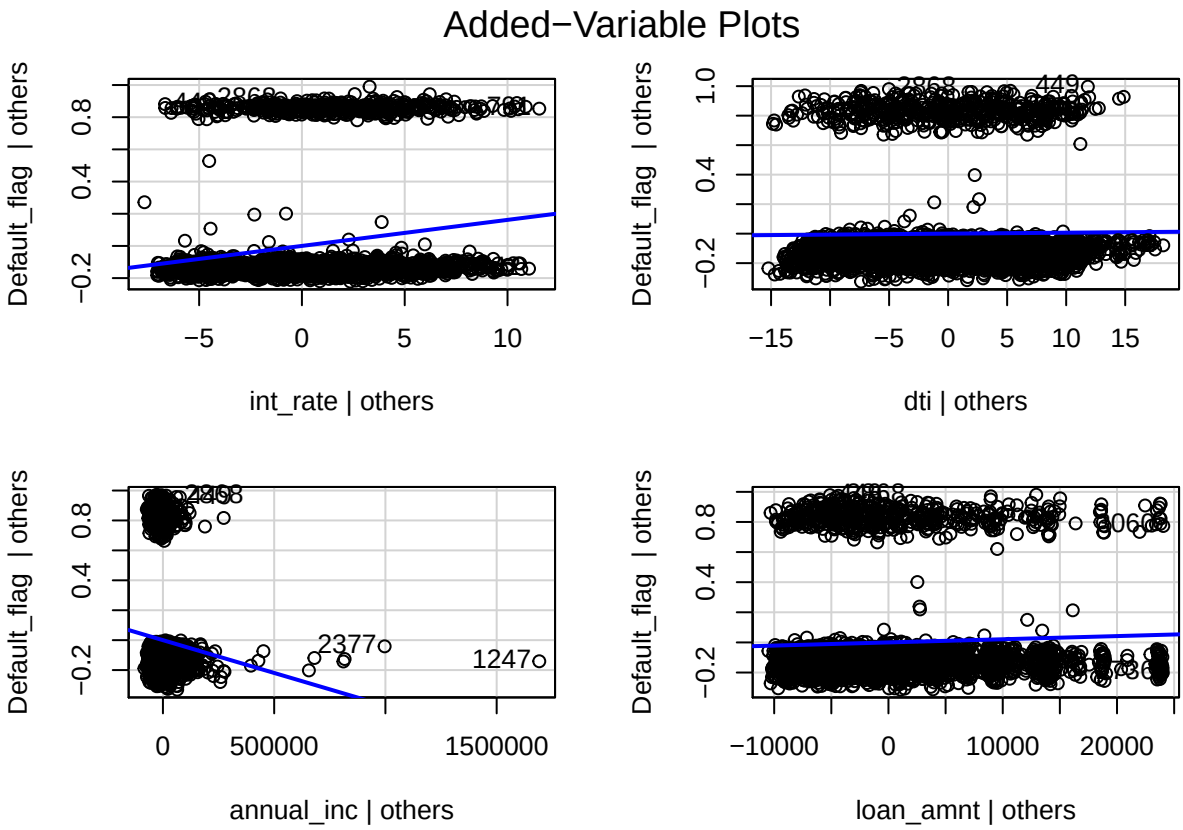
Leverage plots:

```
leveragePlots(fit)
```



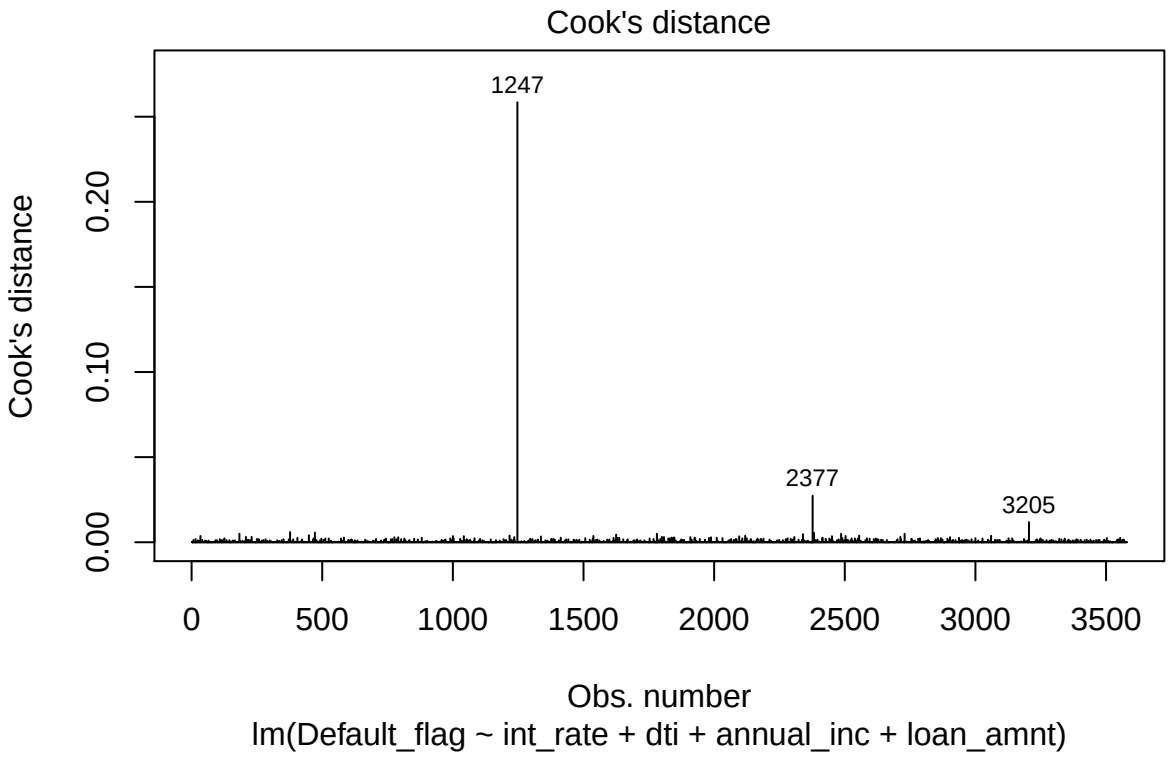
Influential Observations: added variable plots

```
avPlots(fit)
```



Cook's D plot: identify D values > 4/(n-k-1)

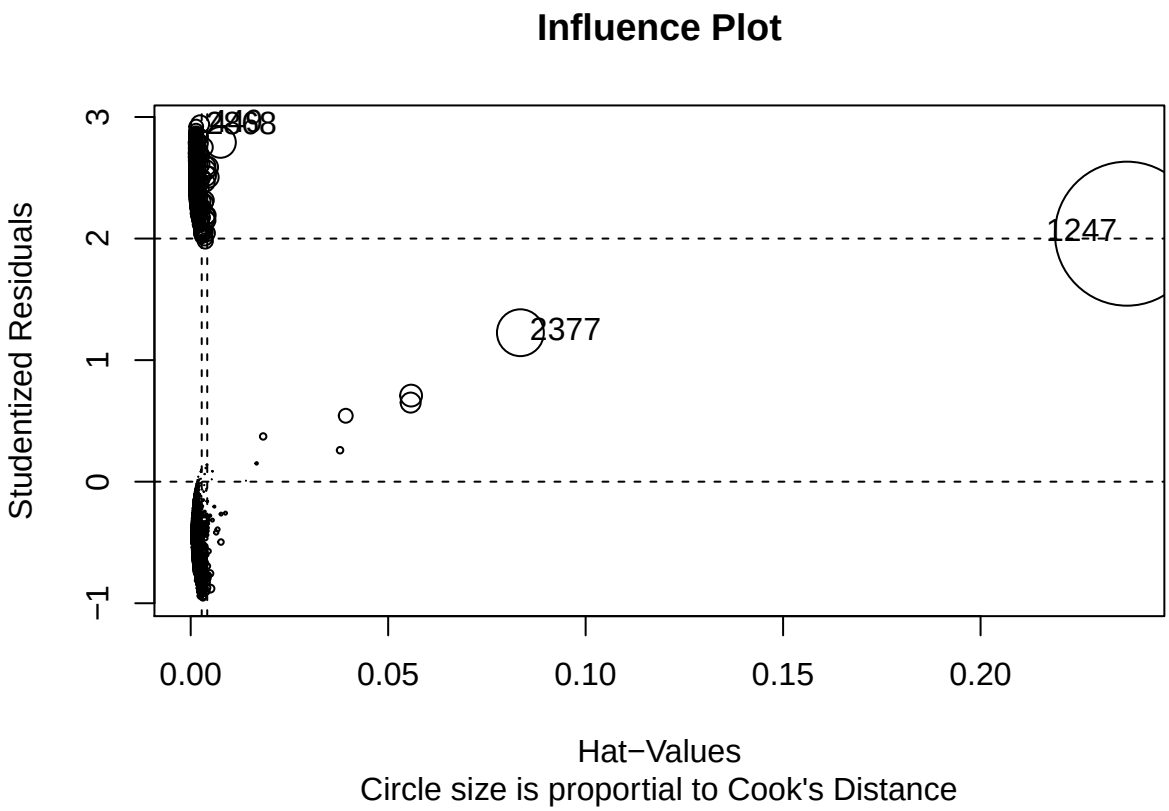
```
cutoff <- 4 / ((nrow(Lend_reg) - length(fit$coefficients) - 2))
plot(fit, which = 4, cook.levels = cutoff)
```



Influence Plot:

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

```
## Warning in plot.window(...): "id.method" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "id.method" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "id.method" is not
## a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "id.method" is not
## a graphical parameter

## Warning in box(...): "id.method" is not a graphical parameter

## Warning in title(...): "id.method" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "id.method" is not a
## graphical parameter
```

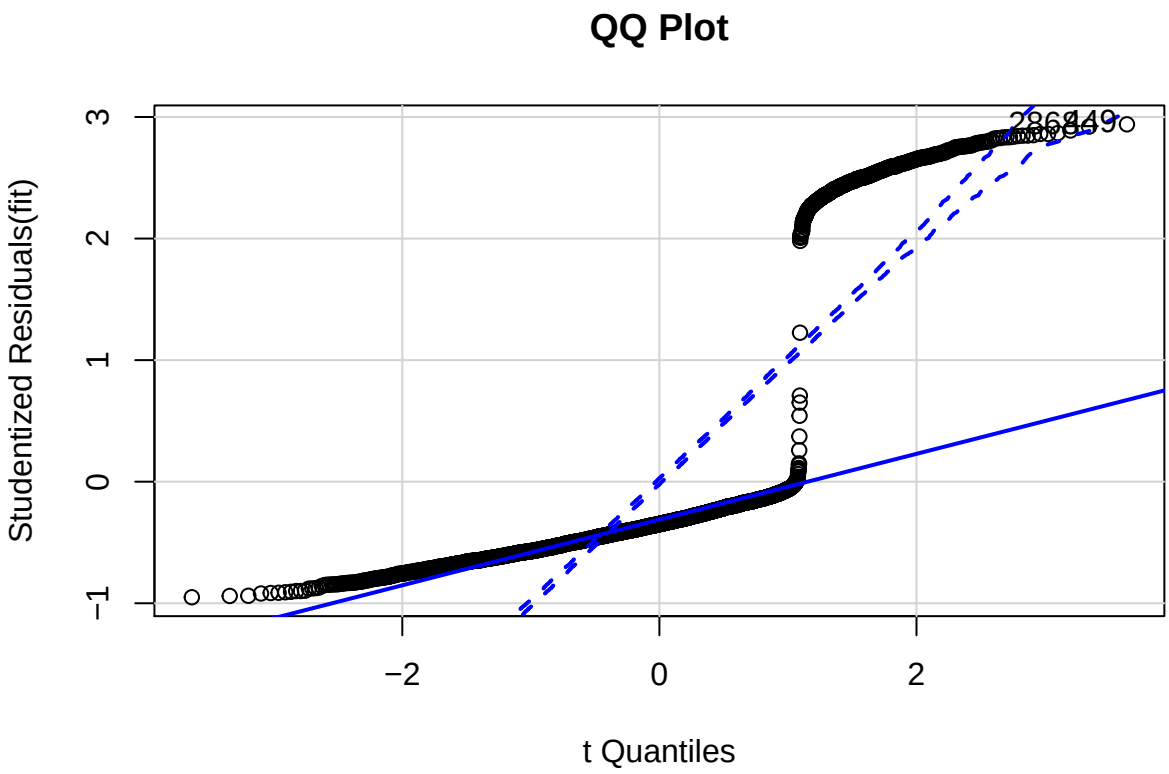



```
##       StudRes         Hat       CookD
## 449  2.939542 0.002367987 0.004093277
## 1247 2.040167 0.237068388 0.258443452
## 2377 1.225478 0.083481181 0.027354385
## 2868 2.921117 0.001392974 0.002375539
```

Normality of Residuals: qq plot for studentized resid

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

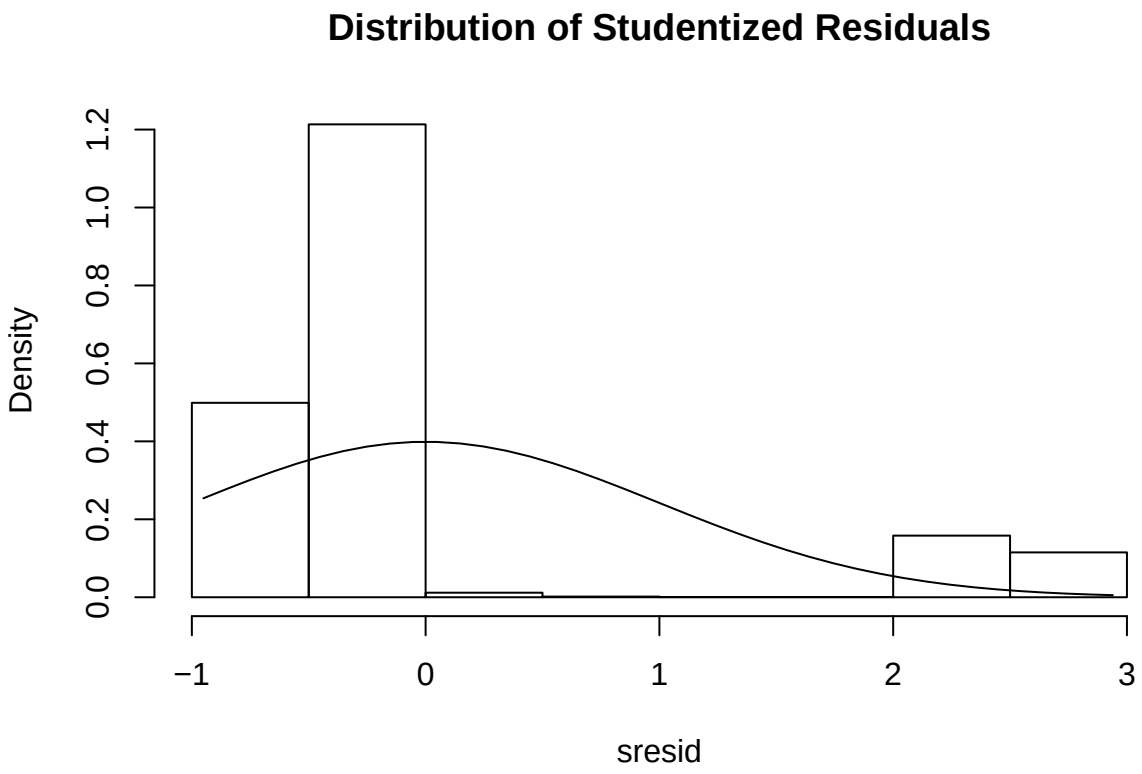
```
## Warning in rlm.default(x, y, weights, method = method, wt.method = wt.method, :
## 'rlm' failed to converge in 20 steps
```




```
## [1]  449 2868
```

Distribution of studentized residuals:

```
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

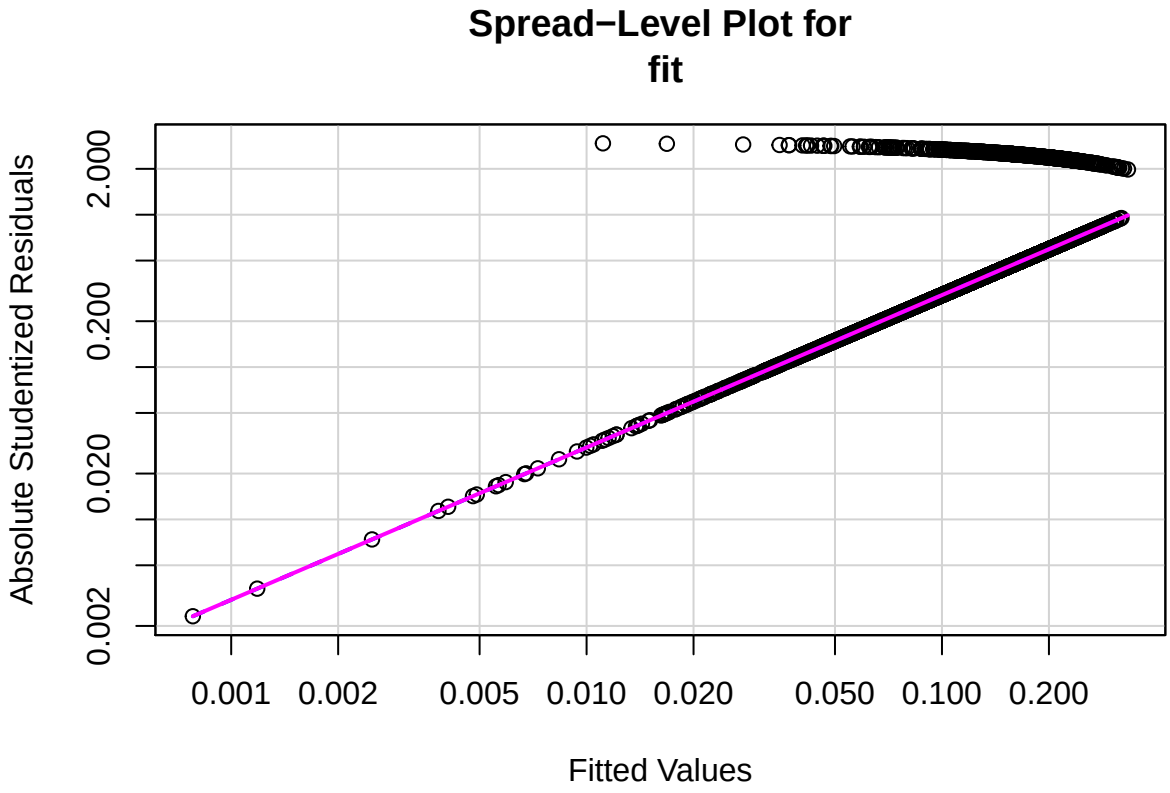
```
ncvTest(fit)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 271.9761, Df = 1, p = < 2.22e-16
```

Plot studentized residuals vs. fitted values

```
spreadLevelPlot(fit)
```

```
## Warning in spreadLevelPlot.lm(fit):
## 26 negative fitted values removed
```




```
##
## Suggested power transformation:  -8.872331e-05
```

Multi-collinearity Evaluate Collinearity

```
vif(fit) # variance inflation factors
```

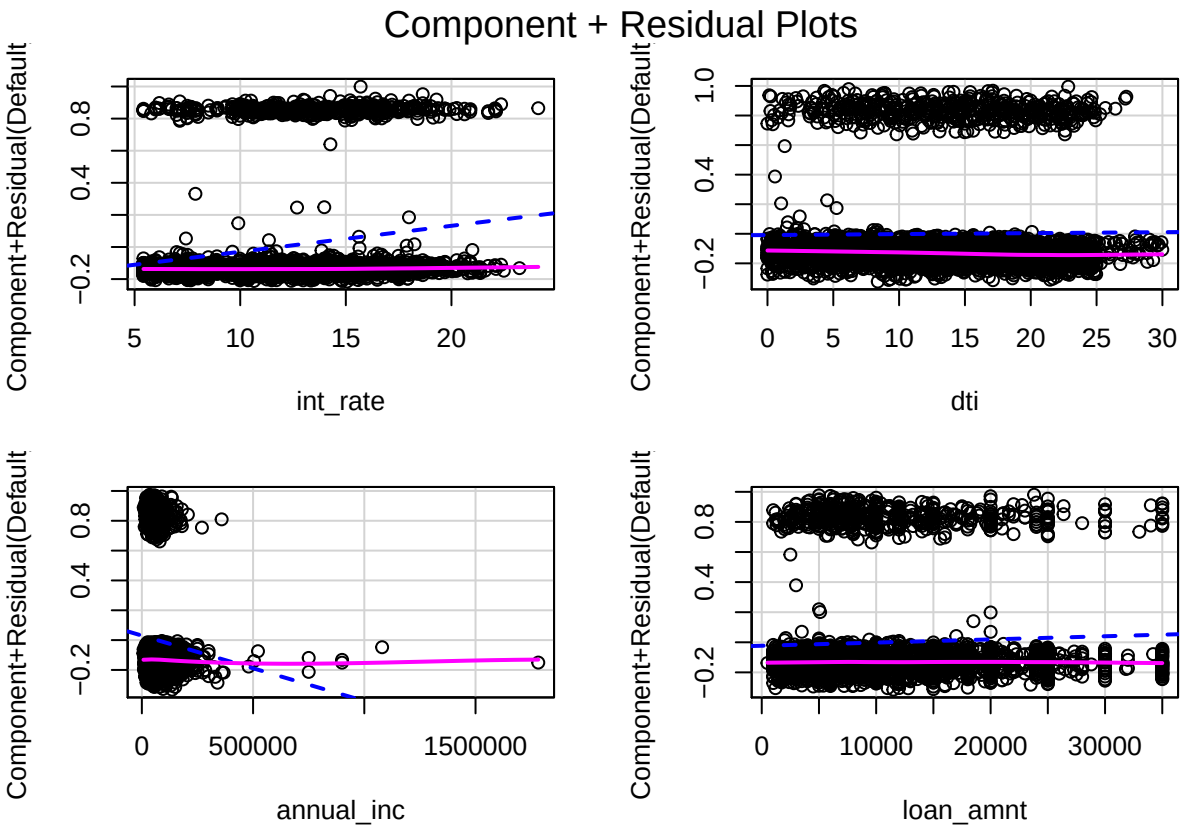
```
##   int_rate        dti annual_inc  loan_amnt
##   1.020642   1.030616   1.025879   1.007472
```

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

```
##   int_rate        dti annual_inc  loan_amnt
##      FALSE      FALSE      FALSE      FALSE
```

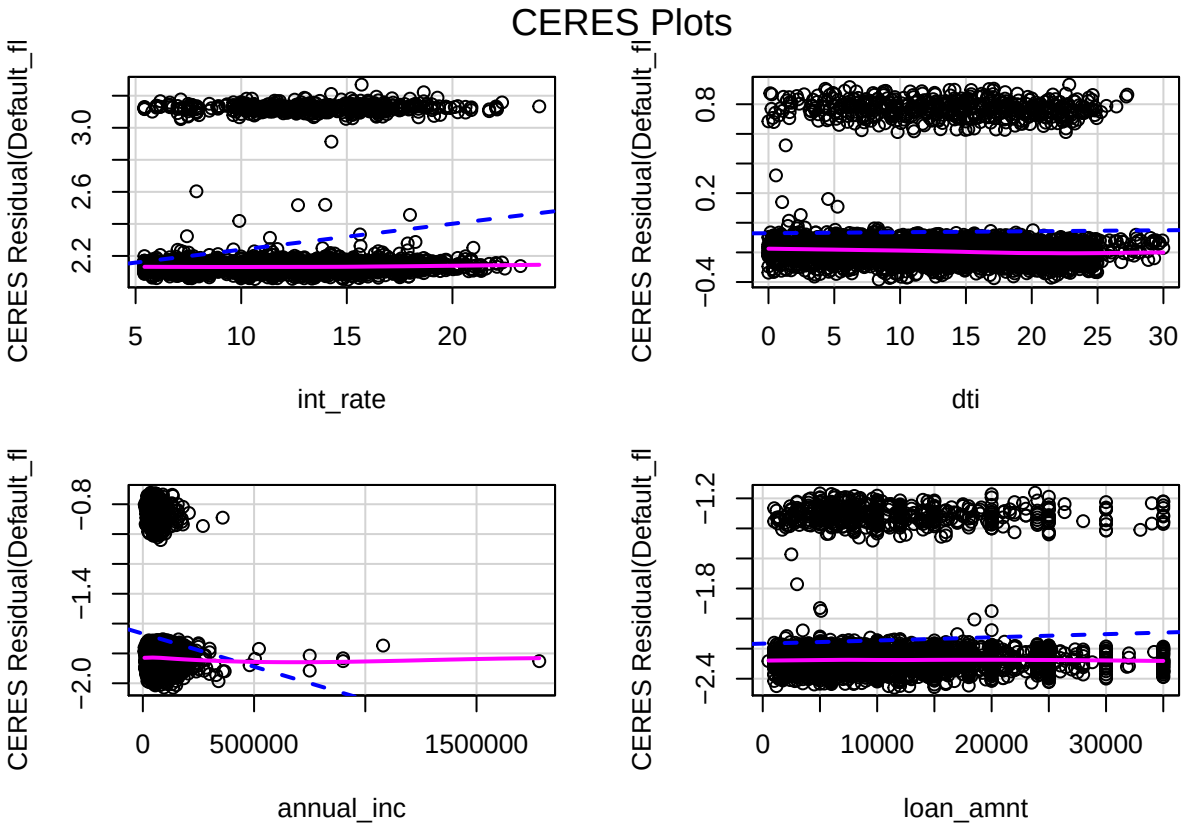
Nonlinearity component + residual plot

```
crPlots(fit)
```



Ceres plots:

```
ceresPlots(fit)
```



Non-independence of Errors Test for Autocorrelated Errors

```
durbinWatsonTest(fit)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1     0.003101623      1.993652   0.864
##  Alternative hypothesis: rho != 0
```

Global test of model assumptions

```
gvmodel <- gvlma(fit)
summary(gvmodel)
```

```
##
## Call:
## lm(formula = Default_flag ~ int_rate + dti + annual_inc + loan_amnt,
##     data = Lend_reg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32002 -0.16833 -0.11749 -0.05753  0.98888
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.611e-02  2.391e-02  -2.347  0.01900 *
```

```
## int_rate     1.618e-02  1.550e-03  10.442  < 2e-16 ***
## dti          6.462e-04  8.537e-04   0.757  0.44912
## annual_inc  -4.385e-07  9.670e-08  -4.535 5.95e-06 ***
## loan_amnt    2.103e-06  7.782e-07   2.702  0.00692 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3372 on 3575 degrees of freedom
## Multiple R-squared:  0.03742,    Adjusted R-squared:  0.03634
## F-statistic: 34.74 on 4 and 3575 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        3175.397 0.000e+00 Assumptions NOT satisfied!
## Skewness           2381.807 0.000e+00 Assumptions NOT satisfied!
## Kurtosis            774.770 0.000e+00 Assumptions NOT satisfied!
## Link Function        17.212 3.344e-05 Assumptions NOT satisfied!
## Heteroscedasticity    1.608 2.047e-01    Assumptions acceptable.
```

```
fit
```

```
##
## Call:
## lm(formula = Default_flag ~ int_rate + dti + annual_inc + loan_amnt,
##     data = Lend_reg)
##
## Coefficients:
## (Intercept)     int_rate          dti   annual_inc    loan_amnt
##  -5.611e-02    1.618e-02    6.462e-04   -4.385e-07    2.103e-06
```

```
summary(fit)
```

```
##
## Call:
## lm(formula = Default_flag ~ int_rate + dti + annual_inc + loan_amnt,
##     data = Lend_reg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32002 -0.16833 -0.11749 -0.05753  0.98888
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.611e-02  2.391e-02  -2.347  0.01900 *
## int_rate     1.618e-02  1.550e-03  10.442  < 2e-16 ***
## dti          6.462e-04  8.537e-04   0.757  0.44912
```

```
## annual_inc  -4.385e-07  9.670e-08  -4.535 5.95e-06 ***
## loan_amnt    2.103e-06  7.782e-07   2.702  0.00692 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3372 on 3575 degrees of freedom
## Multiple R-squared:  0.03742,    Adjusted R-squared:  0.03634
## F-statistic: 34.74 on 4 and 3575 DF,  p-value: < 2.2e-16
```

```
fit1 <- fit
fit2 <- lm(Default_flag~ int_rate + annual_inc + loan_amnt, data = Lend_reg)
```

Compare models:

```
anova(fit1, fit2)
```

```
## Analysis of Variance Table
##
## Model 1: Default_flag ~ int_rate + dti + annual_inc + loan_amnt
## Model 2: Default_flag ~ int_rate + annual_inc + loan_amnt
##   Res.Df    RSS Df Sum of Sq     F Pr(>F)
## 1   3575 406.41
## 2   3576 406.47 -1 -0.065137 0.573 0.4491
```

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

```
## Start:  AIC=-7779.22
## Default_flag ~ int_rate + dti + annual_inc + loan_amnt
##
##              Df Sum of Sq    RSS     AIC
## - dti         1    0.0651 406.47 -7780.7
## <none>                    406.41 -7779.2
## - loan_amnt   1    0.8300 407.24 -7773.9
## - annual_inc  1    2.3381 408.75 -7760.7
## - int_rate    1   12.3957 418.80 -7673.7
##
## Step:  AIC=-7780.65
## Default_flag ~ int_rate + annual_inc + loan_amnt
##
##              Df Sum of Sq    RSS     AIC
## <none>                    406.47 -7780.7
## + dti         1    0.0651 406.41 -7779.2
## - loan_amnt   1    0.8224 407.30 -7775.4
## - annual_inc  1    2.4715 408.94 -7760.9
## - int_rate    1   12.8465 419.32 -7671.3
```

```
step$anova # display results
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
```

```
## Initial Model:
## Default_flag ~ int_rate + dti + annual_inc + loan_amnt
##
## Final Model:
## Default_flag ~ int_rate + annual_inc + loan_amnt
##
##
##    Step Df   Deviance Resid. Df Resid. Dev       AIC
## 1                          3575   406.4078 -7779.224
## 2 - dti  1 0.06513734      3576   406.4730 -7780.650
```

```
leaps <- regsubsets(Default_flag ~ int_rate + dti + No_of_Enquiry +
                      annual_inc + open_acc + loan_amnt,
                      data = Lend_reg, nbest = 10)
```

View results

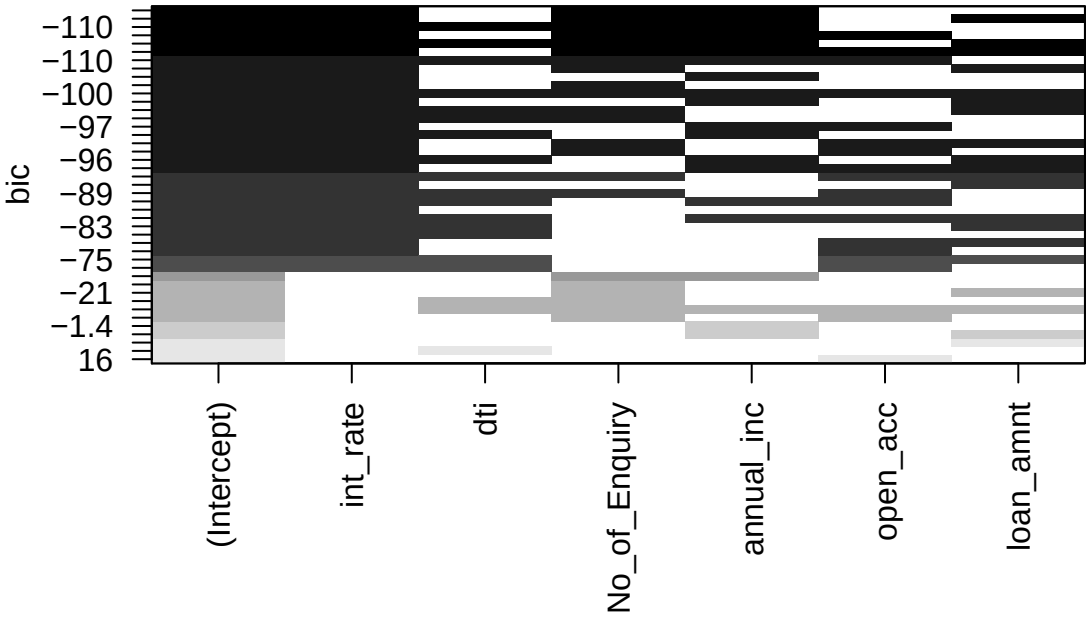
```
summary(leaps)
```

```
## Subset selection object
## Call: regsubsets.formula(Default_flag ~ int_rate + dti + No_of_Enquiry +
##     annual_inc + open_acc + loan_amnt, data = Lend_reg, nbest = 10)
## 6 Variables  (and intercept)
##               Forced in Forced out
## int_rate          FALSE      FALSE
## dti               FALSE      FALSE
## No_of_Enquiry     FALSE      FALSE
## annual_inc        FALSE      FALSE
## open_acc          FALSE      FALSE
## loan_amnt         FALSE      FALSE
## 10 subsets of each size up to 6
## Selection Algorithm: exhaustive
##           int_rate dti No_of_Enquiry annual_inc open_acc loan_amnt
## 1  ( 1 )  "*"      " " " "           " "         " "      " "
## 1  ( 2 )  " "      " " "*"           " "         " "      " "
## 1  ( 3 )  " "      " " " "           "*"         " "      " "
## 1  ( 4 )  " "      " " " "           " "         " "      "*"
## 1  ( 5 )  " "      "*" " "           " "         " "      " "
## 1  ( 6 )  " "      " " " "           " "         "*"      " "
## 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 )  "*"      "*" "*"           " "         " "      " "
```

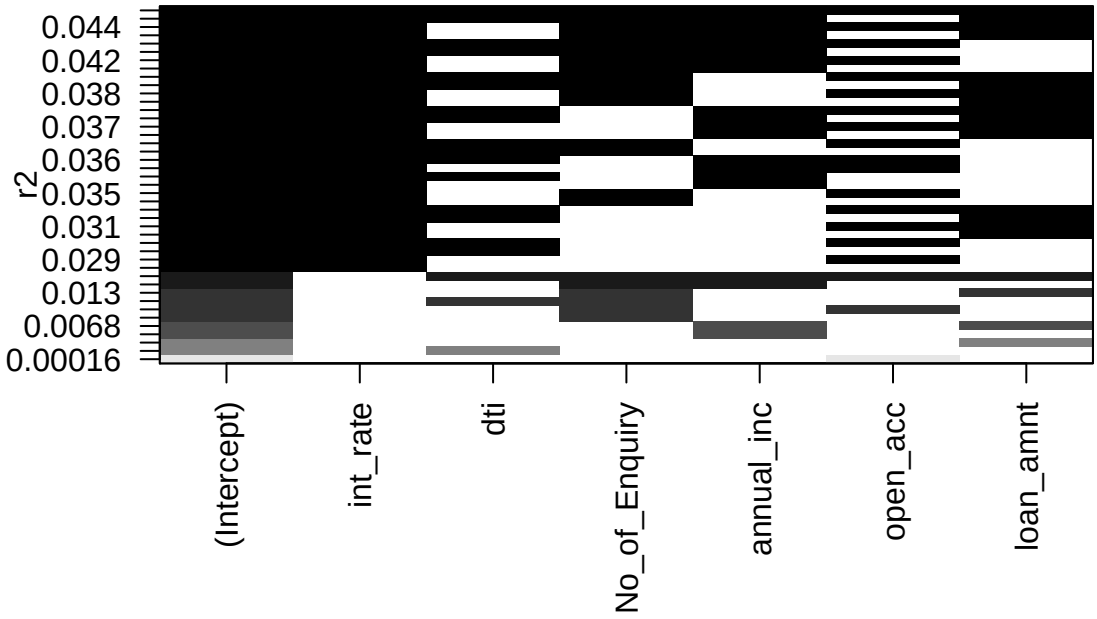
```
## 3  ( 5 )  "*"      " " " " "           "*"        "*"      " "
## 3  ( 6 )  "*"      "*" " " "           "*"        " "      " "
## 3  ( 7 )  "*"      " " "*" "           " "        "*"      " "
## 3  ( 8 )  "*"      "*" " " "           " "        " "      "*"
## 3  ( 9 )  "*"      " " " " "           " "        "*"      "*"
## 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 )  " "      "*" "*" "           "*"        "*"      "*"
## 6  ( 1 )  "*"      "*" "*" "           "*"        "*"      "*"
```

Plot a table of models showing variables in each model. Models are ordered by the selection statistic.

```
plot(leaps)
```

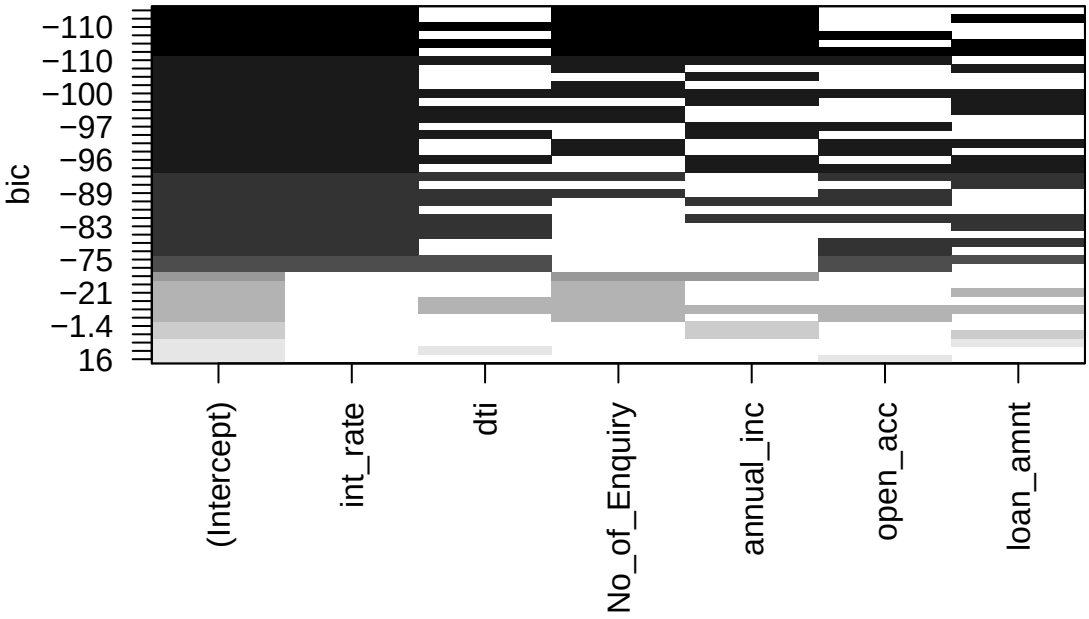



```
plot(leaps, scale = "r2")
```



All Subsets Regression

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




```
summary(leaps)
```

```
## Subset selection object
## Call: regsubsets.formula(Default_flag ~ int_rate + dti + No_of_Enquiry +
##     annual_inc + open_acc + loan_amnt, data = Lend_reg, nbest = 10)
## 6 Variables  (and intercept)
##               Forced in Forced out
## int_rate          FALSE      FALSE
## dti               FALSE      FALSE
## No_of_Enquiry     FALSE      FALSE
## annual_inc        FALSE      FALSE
## open_acc          FALSE      FALSE
## loan_amnt         FALSE      FALSE
## 10 subsets of each size up to 6
## Selection Algorithm: exhaustive
##          int_rate dti No_of_Enquiry annual_inc open_acc loan_amnt
## 1  ( 1 ) "*"      " " " "           " "        " "      " "
## 1  ( 2 ) " "      " " "*"           " "        " "      " "
## 1  ( 3 ) " "      " " " "           "*"        " "      " "
## 1  ( 4 ) " "      " " " "           " "        " "      "*"
## 1  ( 5 ) " "      "*" " "           " "        " "      " "
## 1  ( 6 ) " "      " " " "           " "        "*"      " "
## 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 )  " "      "*" "*"           "*"         "*"       "*"
## 6  ( 1 )  "*"      "*" "*"           "*"         "*"       "*"
```

```
leaps
```

```
## Subset selection object
## Call: regsubsets.formula(Default_flag ~ int_rate + dti + No_of_Enquiry +
##     annual_inc + open_acc + loan_amnt, data = Lend_reg, nbest = 10)
## 6 Variables  (and intercept)
##               Forced in Forced out
## int_rate          FALSE      FALSE
## dti               FALSE      FALSE
## No_of_Enquiry     FALSE      FALSE
## annual_inc        FALSE      FALSE
## open_acc          FALSE      FALSE
## loan_amnt         FALSE      FALSE
## 10 subsets of each size up to 6
## Selection Algorithm: exhaustive
```

```
coef(leaps, 1:6)
```

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

```
## (Intercept)    int_rate
## -0.05130192  0.01587276
##
## [[2]]
##   (Intercept) No_of_Enquiry
##    0.10854262    0.03229894
##
## [[3]]
##   (Intercept)    annual_inc
##  1.649201e-01 -4.098281e-07
##
## [[4]]
##  (Intercept)    loan_amnt
## 1.112258e-01 2.319729e-06
##
## [[5]]
## (Intercept)         dti
## 0.108372107 0.002149522
##
## [[6]]
##  (Intercept)     open_acc
## 0.1276037853 0.0009669877
```

Calculate Relative Importance for Each Predictor

```
calc.relimp(fit, type = c("lmg", "last", "first", "pratt"),
            rela = TRUE)
```

```
## Warning in rev(variances[[p]]) - variances[[p + 1]]: Recycling array of length 1 in vector-array ari
##   Use c() or as.vector() instead.
```

```
## Response variable: Default_flag
## Total response variance: 0.1179677
## Analysis based on 3580 observations
##
## 4 Regressors:
## int_rate dti annual_inc loan_amnt
## Proportion of variance explained by model: 3.74%
## Metrics are normalized to sum to 100% (rela=TRUE).
##
## Relative importance metrics:
##
##                   lmg       last      first      pratt
## int_rate   0.77702692 0.79312465 0.75949602 0.78558530
## dti        0.02446624 0.00416773 0.04633657 0.01413459
## annual_inc 0.13952511 0.14960147 0.13068449 0.14188983
## loan_amnt  0.05898174 0.05310616 0.06348292 0.05839028
##
## Average coefficients for different model sizes:
##
##                         1X           2Xs           3Xs           4Xs
## int_rate    1.587276e-02  1.595281e-02  1.605712e-02  1.618029e-02
## dti         2.149522e-03  1.678455e-03  1.176722e-03  6.462005e-04
```

```
## annual_inc -4.098281e-07 -4.149425e-07 -4.244525e-07 -4.385255e-07
## loan_amnt   2.319729e-06  2.261978e-06  2.193054e-06  2.102746e-06
```

Bootstrap Measures of Relative Importance (1000 samples)

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

Print result

```
booteval.relimp(boot)
```

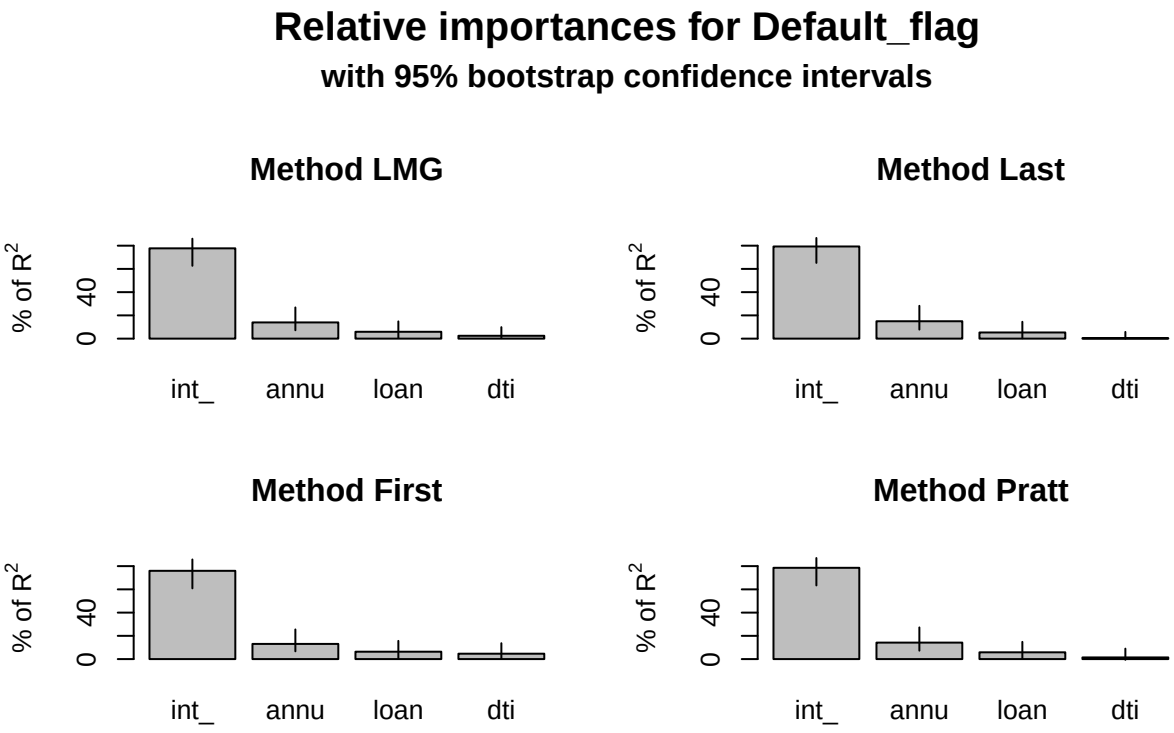
```
## Warning in rev(variances[[p]]) - variances[[p + 1]]: Recycling array of length 1 in vector-array ari
##   Use c() or as.vector() instead.

## Response variable: Default_flag
## Total response variance: 0.1179677
## Analysis based on 3580 observations
##
## 4 Regressors:
## int_rate dti annual_inc loan_amnt
## Proportion of variance explained by model: 3.74%
## Metrics are normalized to sum to 100% (rela=TRUE).
##
## Relative importance metrics:
##
##                   lmg       last      first      pratt
## int_rate   0.77702692 0.79312465 0.75949602 0.78558530
## dti        0.02446624 0.00416773 0.04633657 0.01413459
## annual_inc 0.13952511 0.14960147 0.13068449 0.14188983
## loan_amnt  0.05898174 0.05310616 0.06348292 0.05839028
##
## Average coefficients for different model sizes:
##
##                        1X           2Xs           3Xs           4Xs
## int_rate    1.587276e-02  1.595281e-02  1.605712e-02  1.618029e-02
## dti         2.149522e-03  1.678455e-03  1.176722e-03  6.462005e-04
## annual_inc -4.098281e-07 -4.149425e-07 -4.244525e-07 -4.385255e-07
## loan_amnt   2.319729e-06  2.261978e-06  2.193054e-06  2.102746e-06
##
##
##  Confidence interval information ( 1000 bootstrap replicates, bty= perc ):
## Relative Contributions with confidence intervals:
##
##                                  Lower  Upper
##                  percentage 0.95 0.95    0.95
## int_rate.lmg      0.7770    A___  0.6268  0.8601
## dti.lmg           0.0245    __CD  0.0041  0.0990
## annual_inc.lmg    0.1395    _BC_  0.0720  0.2686
```

```
## loan_amnt.lmg     0.0590    _BCD  0.0053  0.1486
##
## int_rate.last     0.7931    A___  0.6517  0.8663
## dti.last          0.0042    __CD  0.0000  0.0579
## annual_inc.last   0.1496    _BC_  0.0779  0.2824
## loan_amnt.last    0.0531    _BCD  0.0029  0.1450
##
## int_rate.first    0.7595    A___  0.6086  0.8574
## dti.first         0.0463    _BCD  0.0046  0.1368
## annual_inc.first  0.1307    _BC_  0.0667  0.2550
## loan_amnt.first   0.0635    _BCD  0.0075  0.1566
##
## int_rate.pratt    0.7856    A___  0.6338  0.8698
## dti.pratt         0.0141    __CD -0.0063  0.0898
## annual_inc.pratt  0.1419    _BC_  0.0726  0.2727
## loan_amnt.pratt   0.0584    _BCD  0.0045  0.1482
##
## Letters indicate the ranks covered by bootstrap CIs.
## (Rank bootstrap confidence intervals always obtained by percentile method)
## CAUTION: Bootstrap confidence intervals can be somewhat liberal.
##
##
##  Differences between Relative Contributions:
##
##                                          Lower   Upper
##                            difference 0.95 0.95    0.95
## int_rate-dti.lmg            0.7526     *    0.5680  0.8495
## int_rate-annual_inc.lmg     0.6375     *    0.3858  0.7726
## int_rate-loan_amnt.lmg      0.7180     *    0.5006  0.8440
## dti-annual_inc.lmg         -0.1151     *   -0.2504 -0.0119
## dti-loan_amnt.lmg          -0.0345         -0.1318  0.0591
## annual_inc-loan_amnt.lmg    0.0805         -0.0345  0.2342
##
## int_rate-dti.last           0.7890     *    0.6330  0.8624
## int_rate-annual_inc.last    0.6435     *    0.3924  0.7807
## int_rate-loan_amnt.last     0.7400     *    0.5353  0.8496
## dti-annual_inc.last        -0.1454     *   -0.2695 -0.0486
## dti-loan_amnt.last         -0.0489         -0.1329  0.0204
## annual_inc-loan_amnt.last   0.0965         -0.0253  0.2485
##
## int_rate-dti.first          0.7132     *    0.5013  0.8393
## int_rate-annual_inc.first   0.6288     *    0.3792  0.7705
## int_rate-loan_amnt.first    0.6960     *    0.4737  0.8356
## dti-annual_inc.first       -0.0843         -0.2233  0.0229
## dti-loan_amnt.first        -0.0171         -0.1241  0.1005
## annual_inc-loan_amnt.first  0.0672         -0.0442  0.2167
##
## int_rate-dti.pratt          0.7715     *    0.5871  0.8689
## int_rate-annual_inc.pratt   0.6437     *    0.3897  0.7796
## int_rate-loan_amnt.pratt    0.7272     *    0.5072  0.8527
## dti-annual_inc.pratt       -0.1278     *   -0.2655 -0.0206
## dti-loan_amnt.pratt        -0.0443         -0.1405  0.0522
## annual_inc-loan_amnt.pratt  0.0835         -0.0331  0.2394
##
```

```
## * indicates that CI for difference does not include 0.
## CAUTION: Bootstrap confidence intervals can be somewhat liberal.
```

```
plot(booteval.relimp(boot, sort = TRUE)) # plot result
```

```
## Warning in rev(variances[[p]]) - variances[[p + 1]]: Recycling array of length 1 in vector-array ari
##   Use c() or as.vector() instead.
```



$R^2 = 3.74\%$, metrics are normalized to sum 100%.

```
summary(fit)
```

```
##
## Call:
## lm(formula = Default_flag ~ int_rate + dti + annual_inc + loan_amnt,
##     data = Lend_reg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32002 -0.16833 -0.11749 -0.05753  0.98888
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.611e-02  2.391e-02  -2.347  0.01900 *
## int_rate     1.618e-02  1.550e-03  10.442  < 2e-16 ***
## dti          6.462e-04  8.537e-04   0.757  0.44912
```

```
## annual_inc  -4.385e-07  9.670e-08  -4.535 5.95e-06 ***
## loan_amnt    2.103e-06  7.782e-07   2.702  0.00692 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3372 on 3575 degrees of freedom
## Multiple R-squared:  0.03742,    Adjusted R-squared:  0.03634
## F-statistic: 34.74 on 4 and 3575 DF,  p-value: < 2.2e-16
```

```
predict.lm(fit, data.frame(int_rate = 9,
                           dti = 23,
                           annual_inc = 60000,
                           loan_amnt = 150000))
```

```
##         1
## 0.3934775
```

Conclusion:

Given the borrower's interest rate, debt to income ratio, annual income, and loan ammount. The result we get is the probability of a person defaulting the loan. Since we are using regression, we can get a value above 1 and below 0 which we must interpret and defaulter and non defaulter respectively. This is because probabilities cannot be in that range but it is obvious that such extreme values can be easily concluded as defaulters and non defaulters. The challenging task is to classify the borrowers whos predicted value lies between 0 and 1. for which we can set a criteria but for simplicity we set it to 0.5, which means if the value is under 0.5, the borrower is not defaulter and if above 0.5, the borrower is defaulter. Since in our case, we need to minimize risk by minimizing false negatives, accordingly we will set the margin to be even stricter while classifying loan defaulters, because it is not as harmful to predit false positives than to predict false negatives.