Stat 109 Project

May 6, 2019

Daotian Lin, Dhruv Patel, Jason Pease, Scott Schmidt and Henry Steere

I. Motivation

We are a group of people who have a keen interest in financial modeling. In this project, we would like to polish our skills by using a financial dataset. Hence, we chose a dataset published by Lending Club Corporation (Lending Club or LC hereinafter) which contains anonymized data from approved loan applicants.[1] Lending Club manages a peer-to-peer lending platform that matches retail investors and borrowers, in which borrowers can get loans at rates that are more competitive than from a conventional financial institution and lenders can realize proceeds from the underwriting products.

While the loan default rate, FICO score, and other indicators have been well studied by other researchers, in this study, we intend to assess borrowers' underwriting risk by the Debt-to-Income (DTI) ratio. We are interested in the DTI ratio because it is an important indicator of the financial health of individuals. Individuals with a high debt to income ratio may struggle to make repayments on loans and meet other financial obligations, and we believe that DTI ratio will be a useful proxy to determine the borrowers' eligibility.

II. Research Question

We examine predictors of the DTI ratio. Lending Club defines the DTI ratio as "a ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income.[2]

In this project, we build multiple models. We start off building an explanatory model to demonstrate the relationships between DTI and other variables in the dataset. Then, we construct more complex models, as we try to build the best predictive model we can in order to predict the DTI using variables in the dataset.

We believe the explanatory model will be important as it sheds light on the relationship of the variables in a digestible, audience friendly manner. However, we are more concerned with the outcome of the best predictive model. This is because the better a predictive we can build, the more likely it is we can accurately pinpoint an individual's DTI ratio. Although the predictive model can be more difficult to interpret, its importance can not be overstated. With the importance of the machine learning and artifical intelligence growing each day, especially in the realm of finance, it is important that analysts understand how to use these tools and harness them to create better insights for the economy.[3] Through this project, we hope to gain experience and insight in using statistical tools, as well as machine learning techniques, in order to build adequate predictive models.

III. The Dataset

We obtained a dataset from Lending Club containing data from 1.6 million loan applications for the years from 2007 to 2012, of which 748 thousand applications have completed their term and can be used for analysis.

We lack computational resources to handle this volume of data so, for the purpose of this study, we randomly selected a 700 row sample. We also eliminated variables that we were not interested in, in order to reduce the computing power required. A limitation of this dataset is that it may be biased due to it only covering loan applicants that were accepted by Lending Club.

```
lc.sample <- read.csv('LCSample.csv')
```

(i) Selected Variables

The full dataset has over 130 variables, some of which have many missing observations. We inspected the available predictors and decided on a subset that we believed would be related to the DTI ratio. These variables are listed below along with their definitions and our expectations about the effect of these variables.

i. home_ownership

Lending Club defines this as: "a ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income."[2]

This is a categorical variable whose levels are: RENT, OWN, MORTGAGE, OTHER. Because loan repayments are not directly included in the DTI ratio we expect that this variable will indicate the maturity of a borrower. We expect that greater maturity will have a negative effect on DTI ratio.

ii. tax_liens

Lending Club defines this as "number of tax liens,"[2] where a tax lien is a lien imposed by law upon a property to secure the payment of taxes. A tax lien may be imposed for delinquent taxes owed on real property or personal property, or as a result of failure to pay income taxes or other taxes. We include this variable as we believe that it should be an indicator of an individual's personal debt situation. It is expected that an individual with more tax liens would be likely to have higher DTI.

iii. earliest_cr_line

Lending Club defines this as "the month the borrower's earliest reported credit line was opened."[2] We expect this to be related to the age of a borrower. Older borrowers are expected to have more established careers and more secure finances. We expect a longer credit history is a predictor for reduced DTI ratio.

iv. total_acc

Lending Club defines this as "the total number of credit lines currently in the borrower's credit file."[2] We expect that the total number of credit lines currently in the borrower's credit file is directly related to the borrower's payable liabilities and income level which would have a positive effect on DTI ratio.

v. inq_last_6mths

Lending Club defines this as "the number of inquiries in the past 6 months (excluding auto and mortgage inquiries)."[2] This is the number of times that a borrower's credit was checked by a credit bureau, likely due to attempts to apply for credit, in the past six months. We assume this would have a positive effect on an individual's DTI ratio.

vi. tot_coll_amt

Lending Club defines this as "total collection amounts ever owed."[2] We expect that this variable has a positive effect on DTI ratio.

vii. emp_length

Lending Club defines this as "employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years."[2] We expect that this variable correlates with age and maturity of a loan applicant and that longer employment durations correlate with reduced DTI.

viii. addr_state

Lending Club defines this as "the state provided by the borrower in the loan application."[2] We expect that more affluent states will have lower debt to income ratios.

(ii) Data Preparation & Analysis

We decided that some of the variables in the dataset would be more useful after taking steps to preprocess them. In particular, we decided that it would be useful to group states into regions and to work with the difference in time between a borrower's earliest credit line and the date of their loan being issued.

i. Subsetting

This code was used to obtain our train and test subsets from the larger Lending Club data set. `LCAll.csv` is a CSV file made up of Lending Club data for accepted loan applications from 2007-

2012. We include issue_d in our sample to allow us to calculate the time elapsed between an applicants earliest credit line and the Lending Club loan issue date.

```
lc.all <- read.csv('LCAll.csv')
lc.narrow <- lc.all[,c('int_rate','grade','emp_length','home_ownership', 'annual_inc',
    'purpose', 'dti', 'delinq_2yrs', 'earliest_cr_line', 'inq_last_6mths', 'addr_state',
    'pub_rec', 'revol_bal', 'revol_util', 'collections_12_mths_ex_med',
    'application_type', 'acc_now_delinq', 'tot_coll_amt', 'tot_cur_bal',
    'total_rev_hi_lim', 'acc_open_past_24mths', 'avg_cur_bal', 'bc_open_to_buy',
    'bc_util', 'chargeoff_within_12_mths', 'delinq_amnt', 'mo_sin_old_il_acct',
    'mo_sin_old_rev_tl_op', 'mo_sin_rcnt_rev_tl_op', 'mo_sin_rcnt_tl', 'mort_acc',
    'mths_since_recent_bc', 'num_accts_ever_120_pd', 'num_actv_bc_tl', 'num_actv_rev_tl',
    'num_bc_sats', 'num_bc_tl', 'num_il_tl', 'num_op_rev_tl', 'num_rev_accts',
    'num_sats', 'num_tl_120dpd_2m', 'num_tl_30dpd', 'num_tl_90g_dpd_24m',
    'num_tl_op_past_12m', 'pct_tl_nvr_dlq', 'percent_bc_gt_75', 'pub_rec_bankruptcies',
    'tax_liens', 'tot_hi_cred_lim', 'total_bal_ex_mort', 'total_bc_limit',
    'total_il_high_credit_limit', 'issue_d','total_acc')]
lc.filtered <- subset(na.omit(lc.narrow), emp_length != 'n/a')
set.seed(726354)
lc.sample <- lc.filtered[sample(nrow(lc.filtered), 700),]
write.csv(lc.sample, 'LCSample.csv')
```

ii. Grouping States Into Regions

Each loan applicant in the dataset indicates their state of residence. To reduce the number of categorical variables we need to assess in the regression we group states by region as one of North East (NE), Midwest (MW), South (S) and West (W). From the below output, we see that most loan applicants are from the South.

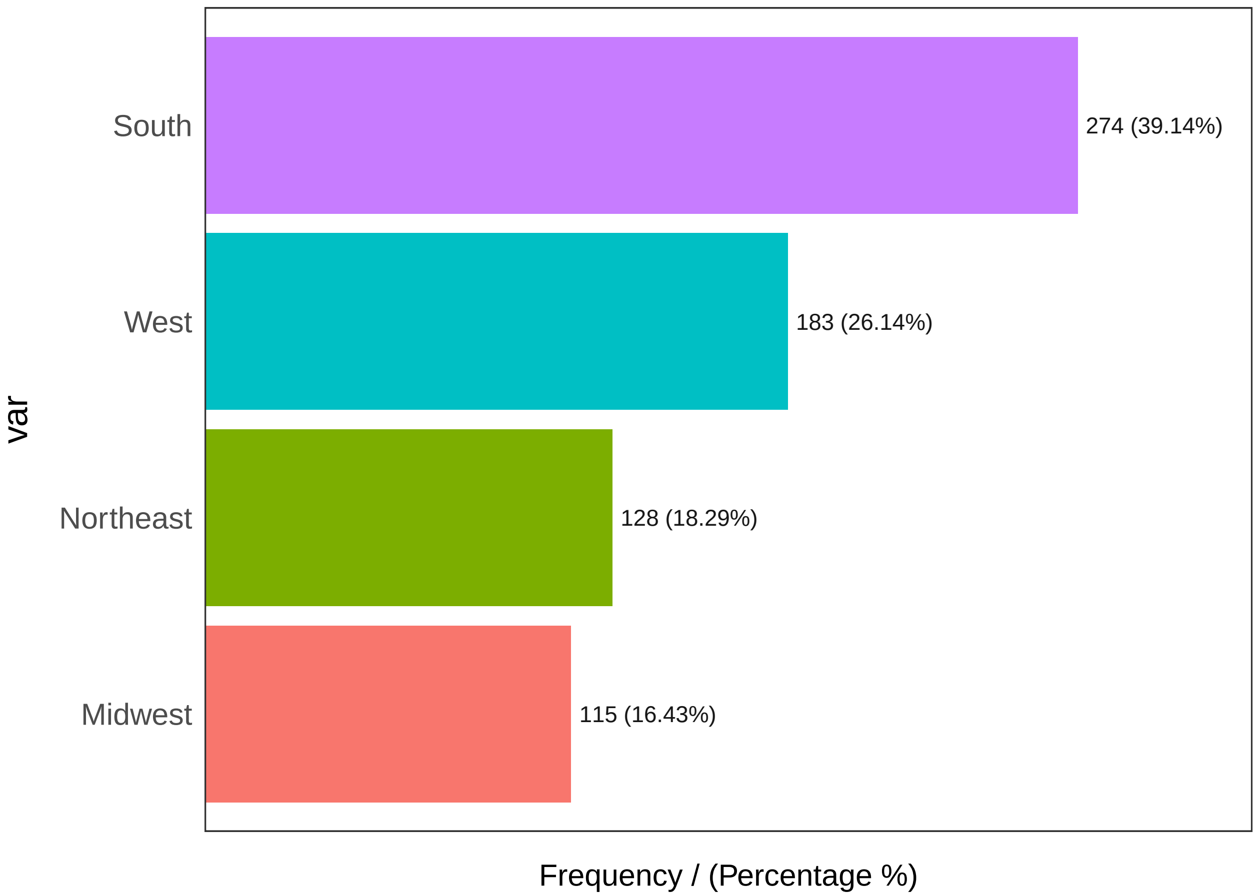
```
NE.name <- c("Connecticut","Maine","Massachusetts","New Hampshire",
             "Rhode Island","Vermont","New Jersey","New York",
             "Pennsylvania")
NE.abrv <- c("CT","ME","MA","NH","RI","VT","NJ","NY","PA")
NE.ref <- c(tolower(NE.name),NE.abrv)

MW.name <- c("Indiana","Illinois","Michigan","Ohio","Wisconsin",
             "Iowa","Kansas","Minnesota","Missouri","Nebraska",
             "North Dakota","South Dakota")
MW.abrv <- c("IN","IL","MI","OH","WI","IA","KS","MN","MO","NE",
             "ND","SD")
MW.ref <- c(tolower(MW.name),MW.abrv)

S.name <- c("Delaware","District of Columbia","Florida","Georgia",
            "Maryland","North Carolina","South Carolina","Virginia",
            "West Virginia","Alabama","Kentucky","Mississippi",
            "Tennessee","Arkansas","Louisiana","Oklahoma","Texas")
S.abrv <- c("DE","DC","FL","GA","MD","NC","SC","VA","WV","AL",
```

```r
            "KY","MS","TN","AR","LA","OK","TX")
S.ref <- c(tolower(S.name),S.abrv)

W.name <- c("Arizona","Colorado","Idaho","New Mexico","Montana",
            "Utah","Nevada","Wyoming","Alaska","California",
            "Hawaii","Oregon","Washington")
W.abrv <- c("AZ","CO","ID","NM","MT","UT","NV","WY","AK","CA",
            "HI","OR","WA")
W.ref <- c(tolower(W.name),W.abrv)

region.list <- list(
  Northeast=NE.ref,
  Midwest=MW.ref,
  South=S.ref,
  West=W.ref)
region.list
```

```
## $Northeast
##  [1] "connecticut"   "maine"         "massachusetts" "new hampshire"
##  [5] "rhode island"  "vermont"       "new jersey"    "new york"
##  [9] "pennsylvania"  "CT"            "ME"            "MA"
## [13] "NH"            "RI"            "VT"            "NJ"
## [17] "NY"            "PA"
##
## $Midwest
##  [1] "indiana"      "illinois"     "michigan"     "ohio"
##  [5] "wisconsin"    "iowa"         "kansas"       "minnesota"
##  [9] "missouri"     "nebraska"     "north dakota" "south dakota"
## [13] "IN"           "IL"           "MI"           "OH"
## [17] "WI"           "IA"           "KS"           "MN"
## [21] "MO"           "NE"           "ND"           "SD"
##
## $South
##  [1] "delaware"             "district of columbia" "florida"
##  [4] "georgia"              "maryland"             "north carolina"
##  [7] "south carolina"       "virginia"             "west virginia"
## [10] "alabama"              "kentucky"             "mississippi"
## [13] "tennessee"            "arkansas"             "louisiana"
## [16] "oklahoma"             "texas"                "DE"
## [19] "DC"                   "FL"                   "GA"
## [22] "MD"                   "NC"                   "SC"
## [25] "VA"                   "WV"                   "AL"
## [28] "KY"                   "MS"                   "TN"
## [31] "AR"                   "LA"                   "OK"
## [34] "TX"
##
## $West
```

```
##  [1] "arizona"    "colorado"   "idaho"      "new mexico" "montana"
##  [6] "utah"       "nevada"     "wyoming"    "alaska"     "california"
## [11] "hawaii"     "oregon"     "washington" "AZ"         "CO"
## [16] "ID"         "NM"         "MT"         "UT"         "NV"
## [21] "WY"         "AK"         "CA"         "HI"         "OR"
## [26] "WA"
```

```r
# CREATE VARIABLE US_REGIONS
lc.sample$us_regions <- sapply(lc.sample$addr_state,
                 function(x) names(region.list)[grep(x,region.list)])
# VIEW THIS TO GET AN IDEA OF HOW LOANS ARE DIVIDED INTO REGIONS
library(funModeling)
freq(lc.sample$us_regions)
```




```
##         var frequency percentage cumulative_perc
## 1     South       274      39.14           39.14
## 2      West       183      26.14           65.28
## 3 Northeast       128      18.29           83.57
## 4   Midwest       115      16.43          100.00
```

iii. Finding the Time Between the Loan Issue Date and Earliest Credit Line

The column containing the earliest credit line (earliest_cr_line) for a loan applicant contained dates represented as month and year of the earliest credit line. We are interested in the difference in time between the earliest credit line taken by a loan applicant and the time that the loan was issued. We converted this date difference to a decimal representing the number of years between an applicant's first credit line and their loan issue date. This variable is interesting because it measures the length of time that an applicant has managing credit and provides information on the age of the applicant, which is not directly available due to privacy concerns on the part of Lending Club.

```
library(lubridate)
library(zoo)
date.to.date.diff <- function(col) {
    date.diff <- difftime(as.yearmon(lc.sample$issue_d,format = "%b-%Y"),
                          as.yearmon(lc.sample[[col]],format = "%b-%Y"),
                  unit = "weeks")/52.25
    as.numeric(date.diff)
}
lc.sample$earliest_cr_line <- date.to.date.diff('earliest_cr_line')
with(lc.sample, summary(earliest_cr_line))
```

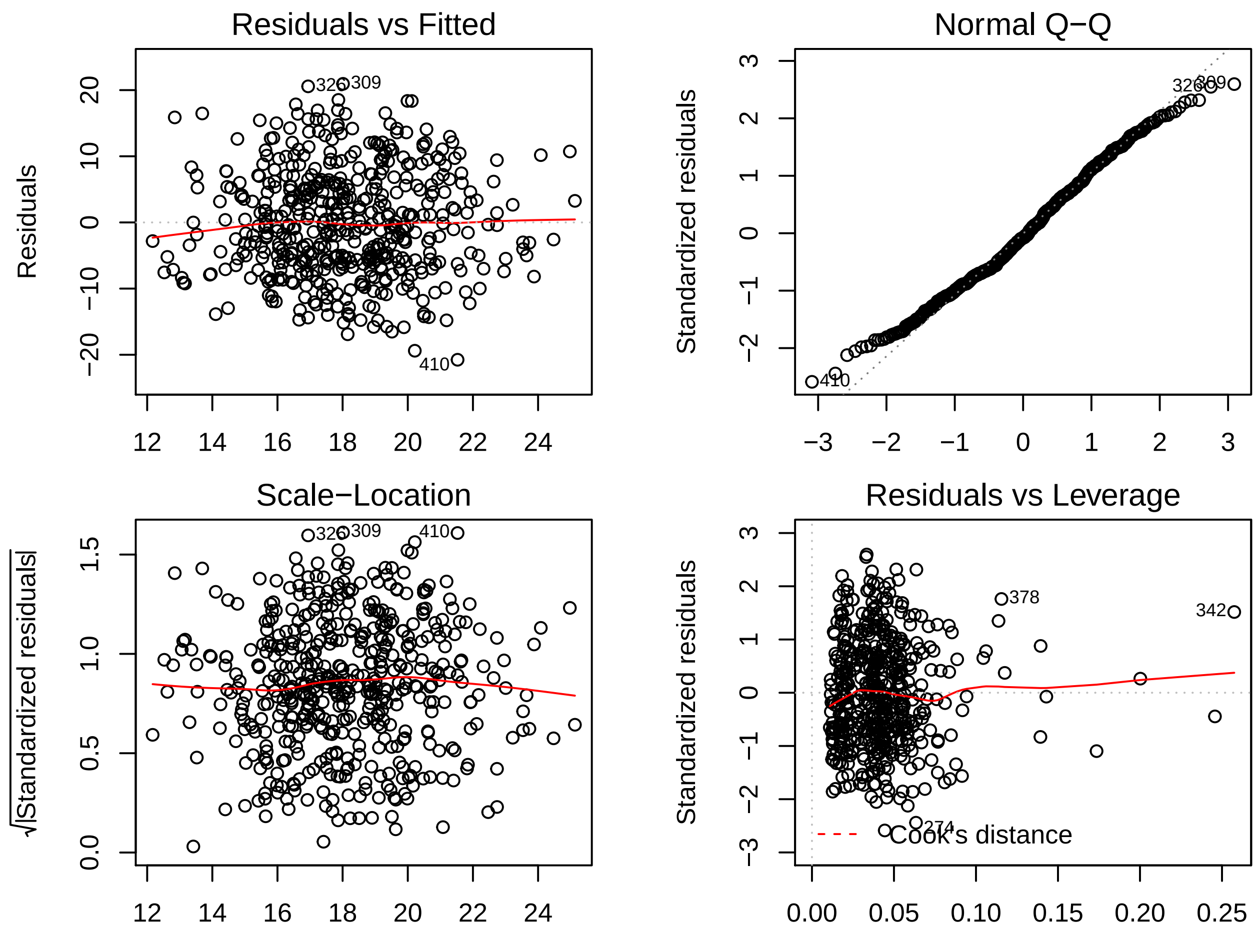
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.248  11.045  14.685  15.922  19.388  45.189
```

IV. The First Model

```
set.seed(21562)
selected <- c('home_ownership','tax_liens','earliest_cr_line',
              'total_acc','inq_last_6mths',
              'tot_coll_amt','dti','emp_length', 'us_regions')
lc.selected <- lc.sample[selected]
train.rows <- sample(nrow(lc.selected), 500)
test.rows <- setdiff(1:700, train.rows)
lc.train <- lc.selected[train.rows,]
lc.test <- lc.selected[test.rows,]
```

Using the subset, we fit all selected variables without any transformation to establish a baseline for comparison with other models.

```
fit <- lm(dti ~ ., data = lc.train)
par(mfrow = c(2, 2), mar = c(2, 4.5, 2, 2))
plot(fit)
```




```r
print(summary(fit))
```

```
##
## Call:
## lm(formula = dti ~ ., data = lc.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.7584  -5.8442  -0.6272   5.7157  20.9483
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        14.9129065  2.0207864   7.380 7.06e-13 ***
## home_ownershipOWN   0.2020738  1.2524514   0.161 0.871892
## home_ownershipRENT  0.3731705  0.8619258   0.433 0.665246
## tax_liens           2.3513202  1.3817970   1.702 0.089472 .
## earliest_cr_line   -0.0687477  0.0584372  -1.176 0.240004
## total_acc           0.1378649  0.0367353   3.753 0.000196 ***
## inq_last_6mths     -0.1547131  0.3864510  -0.400 0.689082
## tot_coll_amt        0.0001249  0.0004565   0.273 0.784590
## emp_length1 year    4.8694040  2.0177879   2.413 0.016186 *
## emp_length10+ years 1.6599016  1.4812725   1.121 0.263023
```

```
## emp_length2 years     3.0858663  1.7840939   1.730 0.084336 .
## emp_length3 years     2.2358444  1.8211591   1.228 0.220161
## emp_length4 years    -1.4213003  1.9083913  -0.745 0.456780
## emp_length5 years     0.2765516  1.9268768   0.144 0.885937
## emp_length6 years     3.9165884  2.3875730   1.640 0.101577
## emp_length7 years     1.8130735  2.0999716   0.863 0.388361
## emp_length8 years     2.6919529  1.9174910   1.404 0.160999
## emp_length9 years     3.3879929  2.2029096   1.538 0.124718
## us_regionsNortheast -2.2951553  1.2654139  -1.814 0.070341 .
## us_regionsSouth     -1.0746124  1.0801821  -0.995 0.320315
## us_regionsWest      -1.5868982  1.1854901  -1.339 0.181336
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.207 on 479 degrees of freedom
## Multiple R-squared:  0.07261,    Adjusted R-squared:  0.03389
## F-statistic: 1.875 on 20 and 479 DF,  p-value: 0.01243
```

The scatter of points in the residuals vs fitted plot appears random, and the loess line is horizontal. The normal Q-Q plot shows a possible departure from normality in the lower quantiles. The residuals vs leverage plot suggests a few observations may be influential.

```r
dti.mean <- mean(lc.train$dti)
dti.mean
```

```
## [1] 18.09266
```

Residual standard error is 8.2072579 which is large given that the mean of the response is 18.09266.

```r
library(car)
ncvTest(fit)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.4159069, Df = 1, p = 0.51899
```

The ncv test fails to reject homoskedasticity.

```r
shapiro.test(residuals(fit))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit)
## W = 0.99028, p-value = 0.002209
```

The Shapiro-Wilk test rejects normality of residuals.

```r
vif(fit)
```

```
##                         GVIF Df GVIF^(1/(2*Df))
## home_ownership      1.264025  2        1.060325
## tax_liens           1.044495  1        1.022005
```

```
## earliest_cr_line 1.224860  1        1.106734
## total_acc        1.184921  1        1.088541
## inq_last_6mths   1.100671  1        1.049129
## tot_coll_amt     1.057714  1        1.028452
## emp_length       1.370135 10        1.015870
## us_regions       1.125413  3        1.019887
```

None of the VIFs is above 10 suggesting that multicollinearity is not a problem for this model.

```
fit.cooks.distance <- cooks.distance(fit)
fit.cooks.distance[which(fit.cooks.distance > 4 / nrow(lc.train))]
```

```
##         197         566         612         569         630         410
## 0.011740401 0.013847950 0.017321241 0.011131287 0.008190579 0.014801346
##         361          27         262         308         596         206
## 0.008277371 0.013313235 0.010637858 0.008908051 0.009591745 0.010797555
##         428         329         507         213         342         684
## 0.008281963 0.011905146 0.009861090 0.008641625 0.037970561 0.009390854
##         481         254         205         326         418         458
## 0.012033552 0.011494193 0.008413224 0.010555458 0.011528508 0.011934722
##         378         309         274         101
## 0.019293507 0.011082037 0.019230899 0.008829336
```

There are 28 observations with a large Cook's distance. This is 5.6% of observations in the dataset and is not alarming. Transformations that reduce extreme values may help in addressing this.

```
k <- length(selected) - 1
fit.hatvalues <- hatvalues(fit)
large.hat.values <- sort(fit.hatvalues[which(fit.hatvalues > 3 * (k + 1) / nrow(lc.train))],
                         decreasing = TRUE)
print(large.hat.values)
```

```
##        342        109        195        481        259        685
## 0.25747069 0.24570342 0.20021601 0.17352507 0.14290367 0.13942256
##        130         80        378        569        375        158
## 0.13930171 0.11753009 0.11553565 0.11376248 0.10614399 0.10445653
##         41        651        197        350        361        135
## 0.09426396 0.09173894 0.09147722 0.08853614 0.08784668 0.08533339
##        532        254        351         56         70        329
## 0.08477310 0.08414724 0.08368032 0.08348592 0.08111646 0.08088832
##        487         45        231        308        564        552
## 0.07852228 0.07686587 0.07677912 0.07670891 0.07645590 0.07600525
##        480        625        160        224        171        581
## 0.07414160 0.07290533 0.07269612 0.07210823 0.07193936 0.07107554
##        265        418        212        600         60        210
## 0.07002387 0.06886507 0.06860491 0.06802522 0.06775923 0.06720122
##        590        187        370        486        529        222
## 0.06679814 0.06674282 0.06673228 0.06582741 0.06557953 0.06544468
##        207        379        535        612         72        274
## 0.06511699 0.06499261 0.06395404 0.06366408 0.06348767 0.06342889
```

```
##        646        622        206        304         43          5
## 0.06341625 0.06267966 0.06142145 0.06141447 0.06130835 0.06082424
##         87         33        376        554        456        494
## 0.06063538 0.06045662 0.06032508 0.06016849 0.05992526 0.05981698
##        543        642        280        377        414        610
## 0.05940271 0.05915833 0.05903336 0.05891075 0.05864966 0.05849760
##         27        337        181         50        209        567
## 0.05842209 0.05803909 0.05782398 0.05780971 0.05697131 0.05693545
##        436        294        541        457        652        444
## 0.05650061 0.05646316 0.05626365 0.05608741 0.05606086 0.05604520
##        103        697        596        523        316        400
## 0.05602100 0.05593030 0.05532338 0.05519244 0.05508570 0.05500982
##        201        422        261        668        290        268
## 0.05496561 0.05491432 0.05479114 0.05478634 0.05473874 0.05472075
##        637        607        356        374        173        626
## 0.05464214 0.05455862 0.05450052 0.05449150 0.05432155 0.05422044
##        647
## 0.05413301
```

```r
length(large.hat.values)
```

```
## [1] 103
```

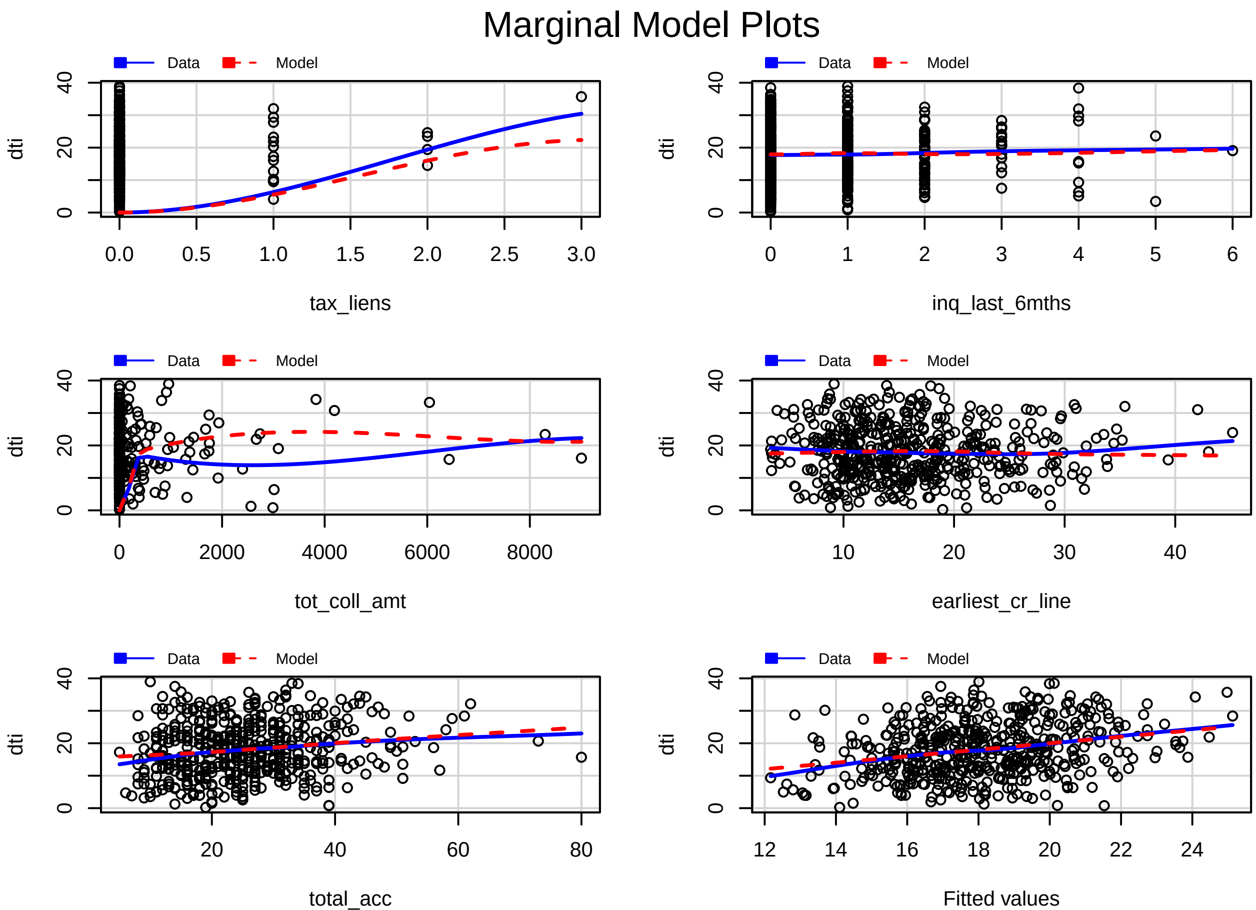
A large number of observations have large hat values. This is possibly due to skew in a predictor or response and may be remedied by a transformation.

```r
library(lmtest)
resettest(fit)
```

```
##
##  RESET test
##
## data:  fit
## RESET = 0.89531, df1 = 2, df2 = 477, p-value = 0.4092
```

The reset test fails to reject the null hypothesis that no transformations of the model are required.

```r
mmps(fit, terms = ~tax_liens + inq_last_6mths + tot_coll_amt + earliest_cr_line + total_acc)
```

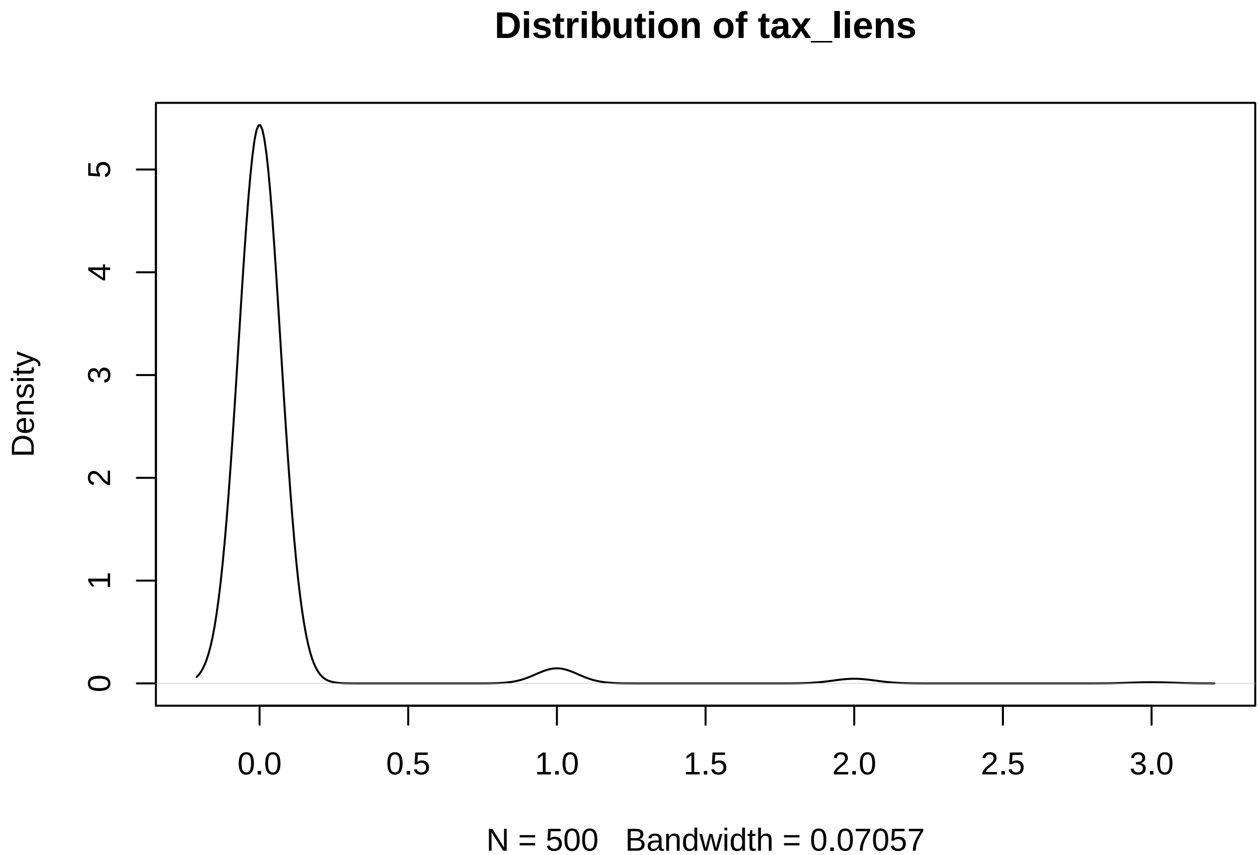


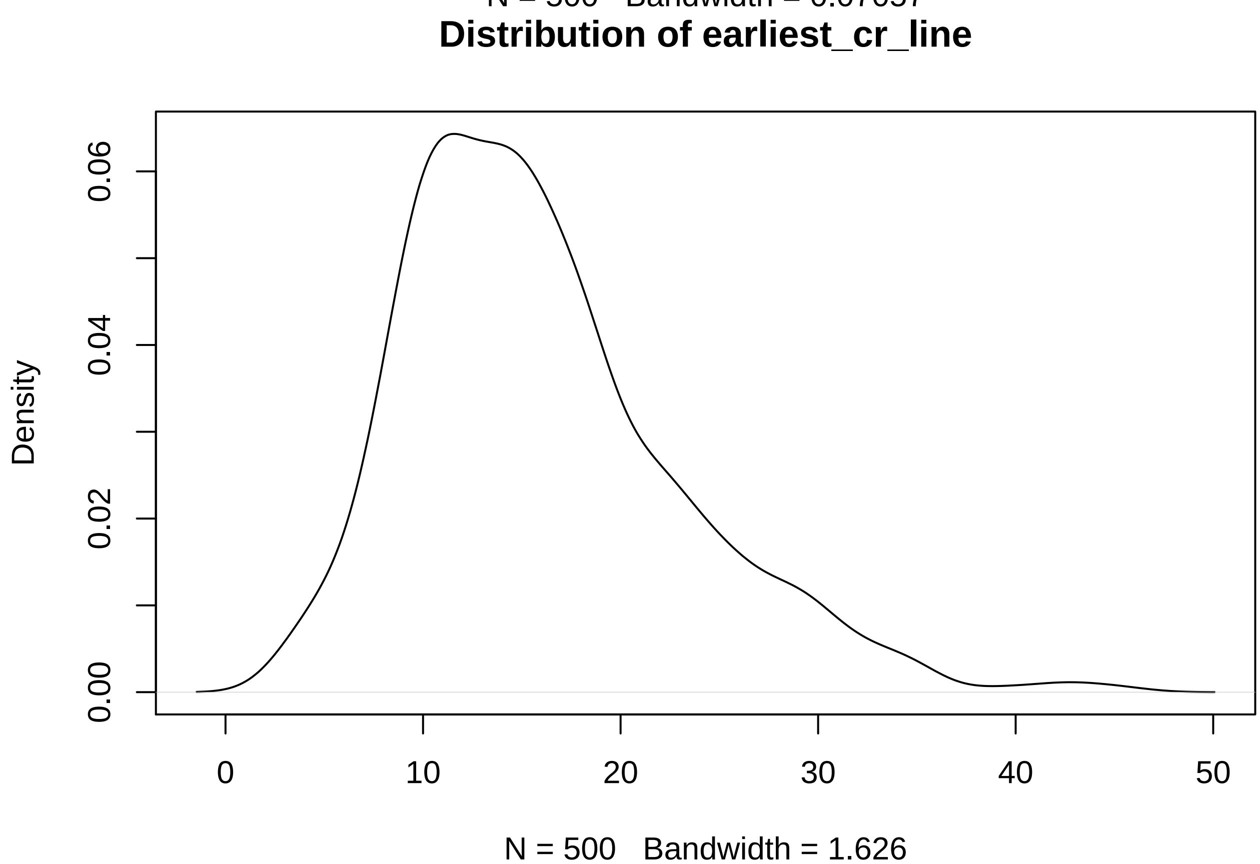
However, marginal model plots suggest that the functional form of the model is wrong.

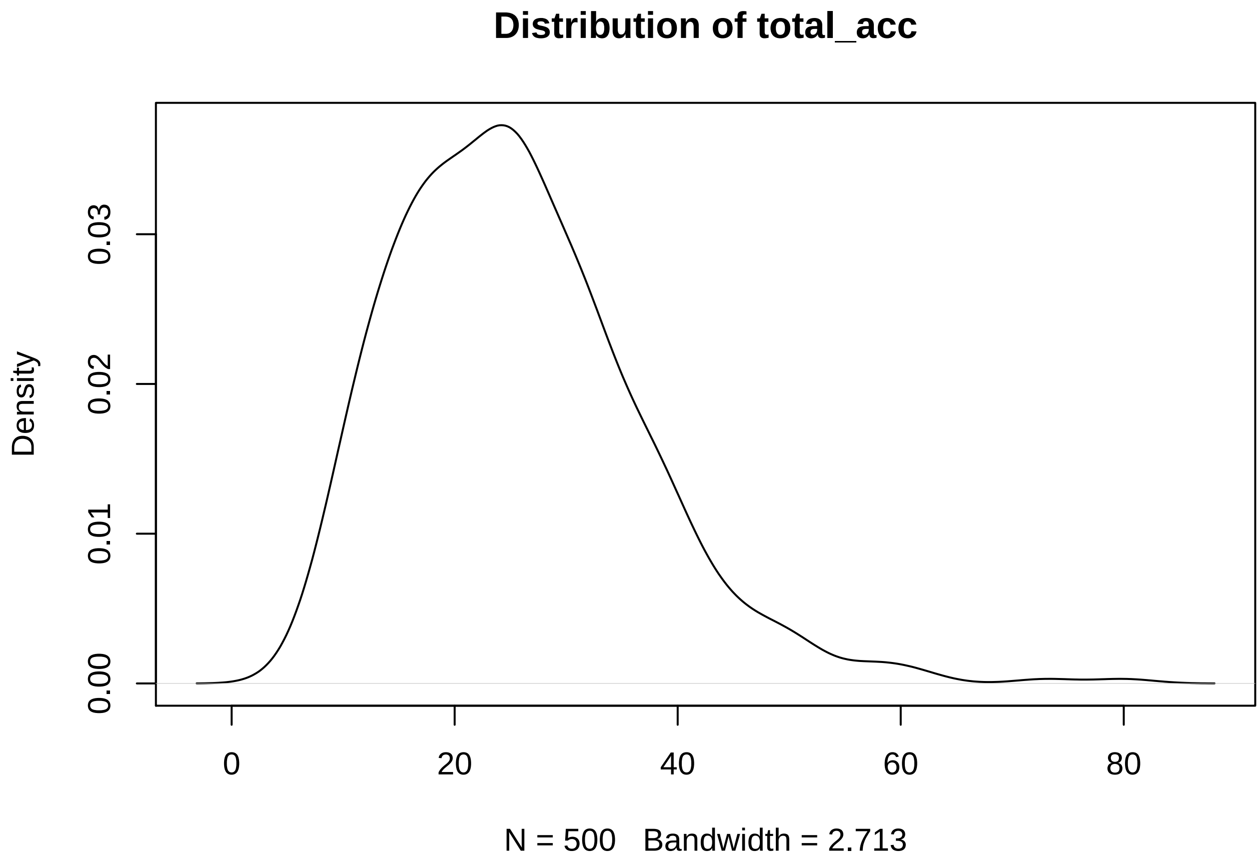
Manual Transformation

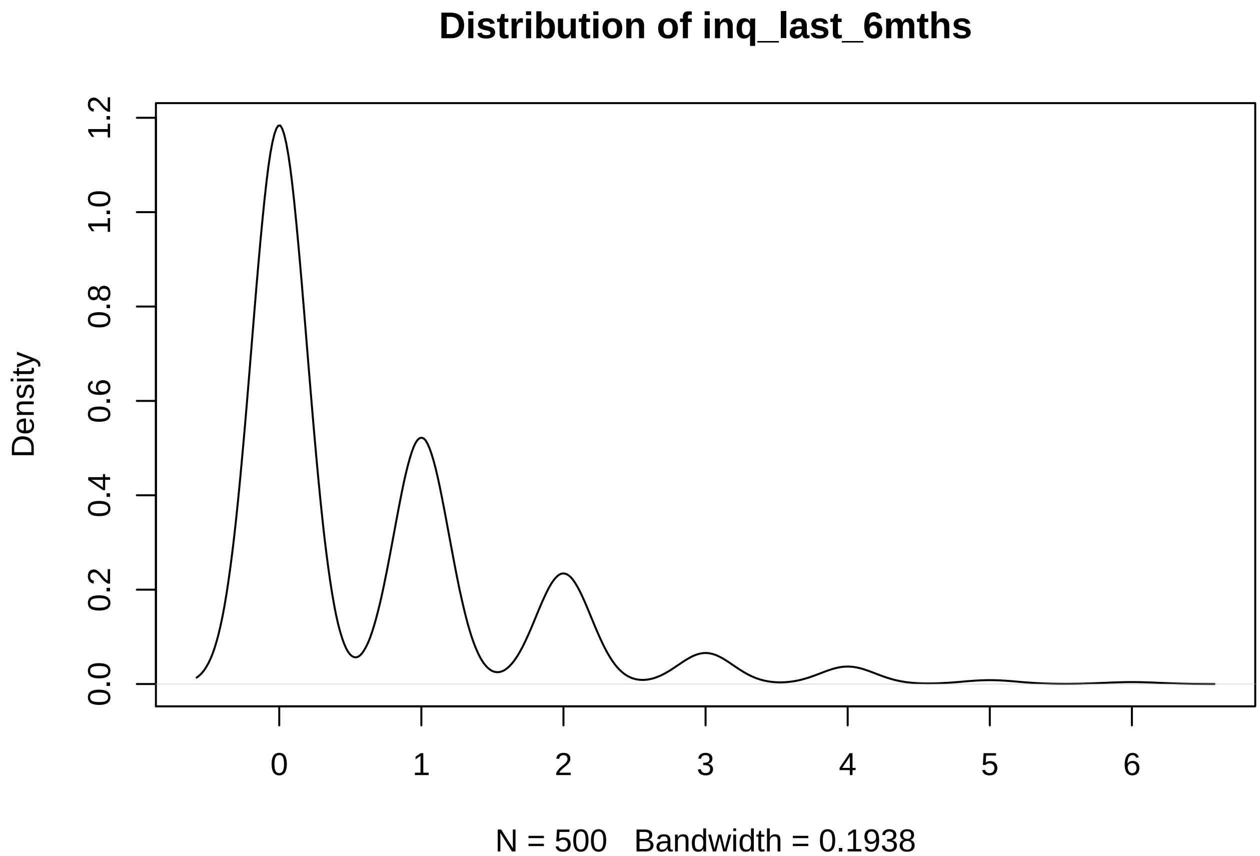
The distribution of each numeric predictor is plotted below. This may suggest transformations toward normality to improve the fit of the model.

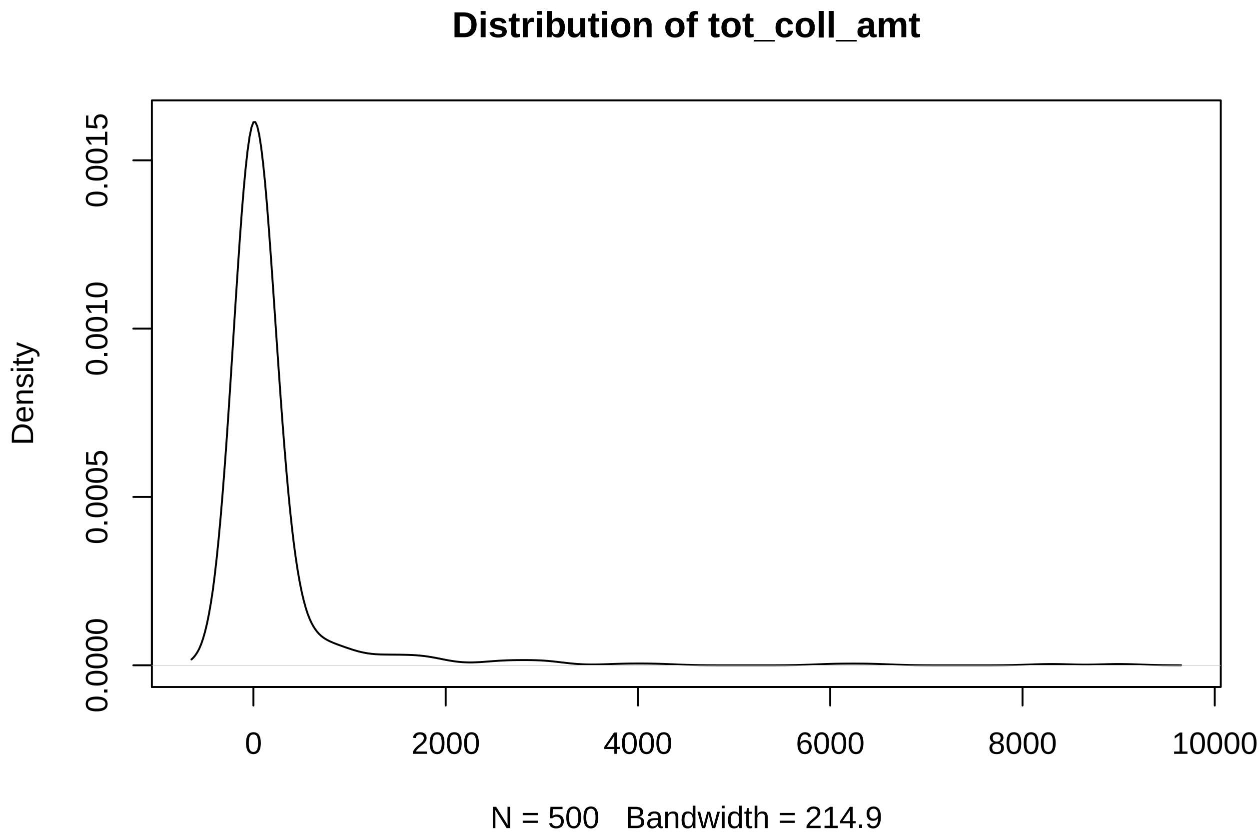
```
numeric.variables <- c('tax_liens','earliest_cr_line',
                       'total_acc','inq_last_6mths',
                       'tot_coll_amt')
for (col in numeric.variables) {
    plot(density(lc.train[[col]]), main = paste("Distribution of", col))
}
```


Distribution of tax_liens
Density
0
1
2
3
4
5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
N = 500   Bandwidth = 0.07057


Distribution of earliest_cr_line
Density
0.00
0.02
0.04
0.06
0
10
20
30
40
50
N = 500   Bandwidth = 1.626


Distribution of total_acc
Density
0.00
0.01
0.02
0.03
0
20
40
60
80
N = 500 Bandwidth = 2.713


Distribution of inq_last_6mths
Density
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
1
2
3
4
5
6
N = 500 Bandwidth = 0.1938



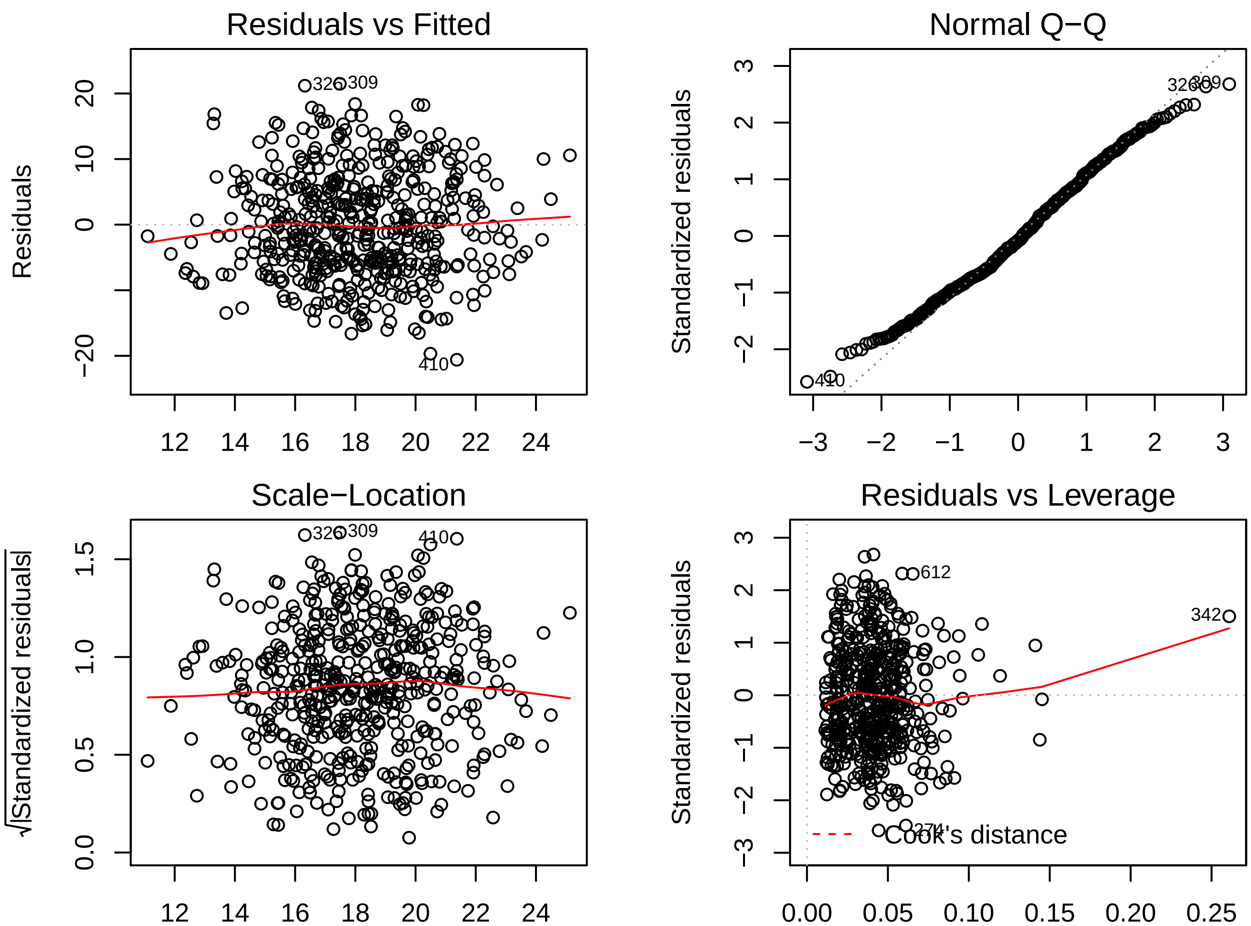
Apply log transformations to earliest_cr_line and tot_coll_amt and apply a square root transformation to total_acc, to bring them closer to normality and then fit the model again.

```
fit.transformed <- lm(dti ~ home_ownership + tax_liens +
                          inq_last_6mths + log(1+tot_coll_amt)
                          + log(earliest_cr_line) +
                          emp_length + us_regions +
                          sqrt(total_acc), data = lc.train)
print(summary(fit.transformed))
```

```
##
## Call:
## lm(formula = dti ~ home_ownership + tax_liens + inq_last_6mths +
##     log(1 + tot_coll_amt) + log(earliest_cr_line) + emp_length +
##     us_regions + sqrt(total_acc), data = lc.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.6007  -5.9050  -0.6563   5.7774  21.4725
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          14.082484   3.101770   4.540 7.12e-06 ***
## home_ownershipOWN     0.281895   1.250232   0.225   0.8217
## home_ownershipRENT    0.364531   0.857930   0.425   0.6711
```

```
## tax_liens              2.380843   1.382903   1.722   0.0858 .
## inq_last_6mths        -0.137858   0.385894  -0.357   0.7211
## log(1 + tot_coll_amt)  0.003613   0.150166   0.024   0.9808
## log(earliest_cr_line) -1.746249   0.935092  -1.867   0.0624 .
## emp_length1 year       4.782033   2.013036   2.376   0.0179 *
## emp_length10+ years    1.725566   1.474366   1.170   0.2424
## emp_length2 years      3.002512   1.780037   1.687   0.0923 .
## emp_length3 years      2.179066   1.815628   1.200   0.2307
## emp_length4 years     -1.525716   1.907542  -0.800   0.4242
## emp_length5 years      0.223765   1.919581   0.117   0.9072
## emp_length6 years      3.754006   2.387192   1.573   0.1165
## emp_length7 years      1.701496   2.095649   0.812   0.4172
## emp_length8 years      2.633827   1.912260   1.377   0.1691
## emp_length9 years      3.297011   2.199333   1.499   0.1345
## us_regionsNortheast   -2.202727   1.263670  -1.743   0.0820 .
## us_regionsSouth       -1.037196   1.074906  -0.965   0.3351
## us_regionsWest        -1.489423   1.180625  -1.262   0.2077
## sqrt(total_acc)        1.598437   0.383239   4.171 3.60e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.183 on 479 degrees of freedom
## Multiple R-squared:  0.07817,    Adjusted R-squared:  0.03968
## F-statistic: 2.031 on 20 and 479 DF,  p-value: 0.005431
```

```r
par(mfrow = c(2, 2), mar = c(2, 4.5, 2, 2))
plot(fit.transformed)
```



After transformation adjusted R-squared has improved.

```r
ncvTest(fit.transformed)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.3008234, Df = 1, p = 0.58337
```

The ncv test fails to reject homoskedasticity.

```r
shapiro.test(residuals(fit.transformed))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit.transformed)
## W = 0.99091, p-value = 0.003617
```

The Shapiro-Wilk test rejects normality of residuals so this model is invalid.

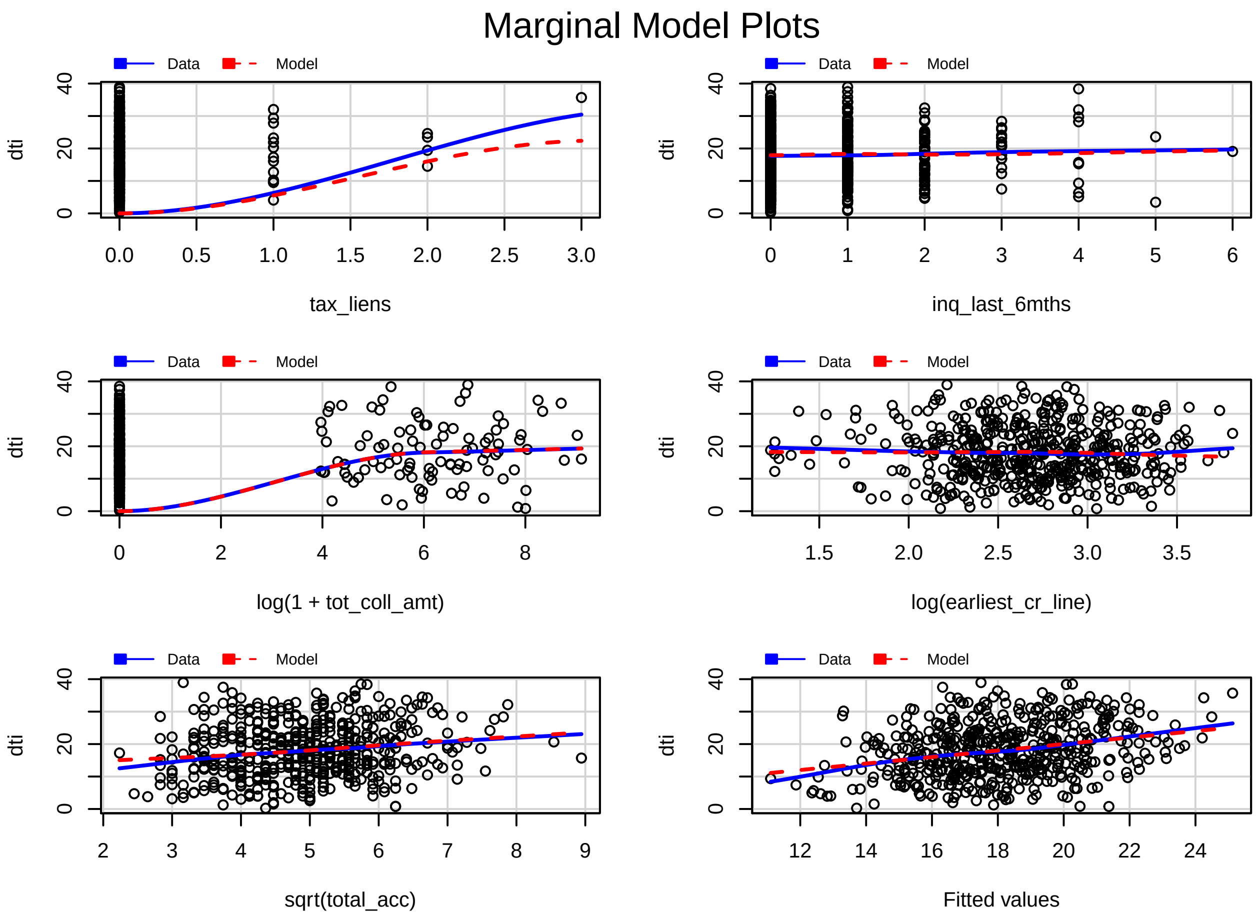
```r
resettest(fit.transformed)
```

```
##
##  RESET test
##
```

```
## data:  fit.transformed
## RESET = 1.548, df1 = 2, df2 = 477, p-value = 0.2137
```

The reset test fails to reject the hypothesis that no transformation is required.

```r
mmps(fit.transformed, terms = ~ tax_liens +
                         inq_last_6mths +
                         log(1+tot_coll_amt) +
                         log(earliest_cr_line) +
                         sqrt(total_acc))
```



Marginal model plots show a better fit, but there is a discrepancy in the fit for `tax_liens`.

Box-Cox Transformation

We use the Box-Cox transformation to transform the response to normality and apply the manually selected transformations of predictors from above.

```r
powerTransform(dti ~ home_ownership + tax_liens +
                   inq_last_6mths + log(1+tot_coll_amt) + log(earliest_cr_line) +
                   emp_length + us_regions +
                   sqrt(total_acc), data = lc.train)
```

```
## Estimated transformation parameter
##        Y1
## 0.7458105
```

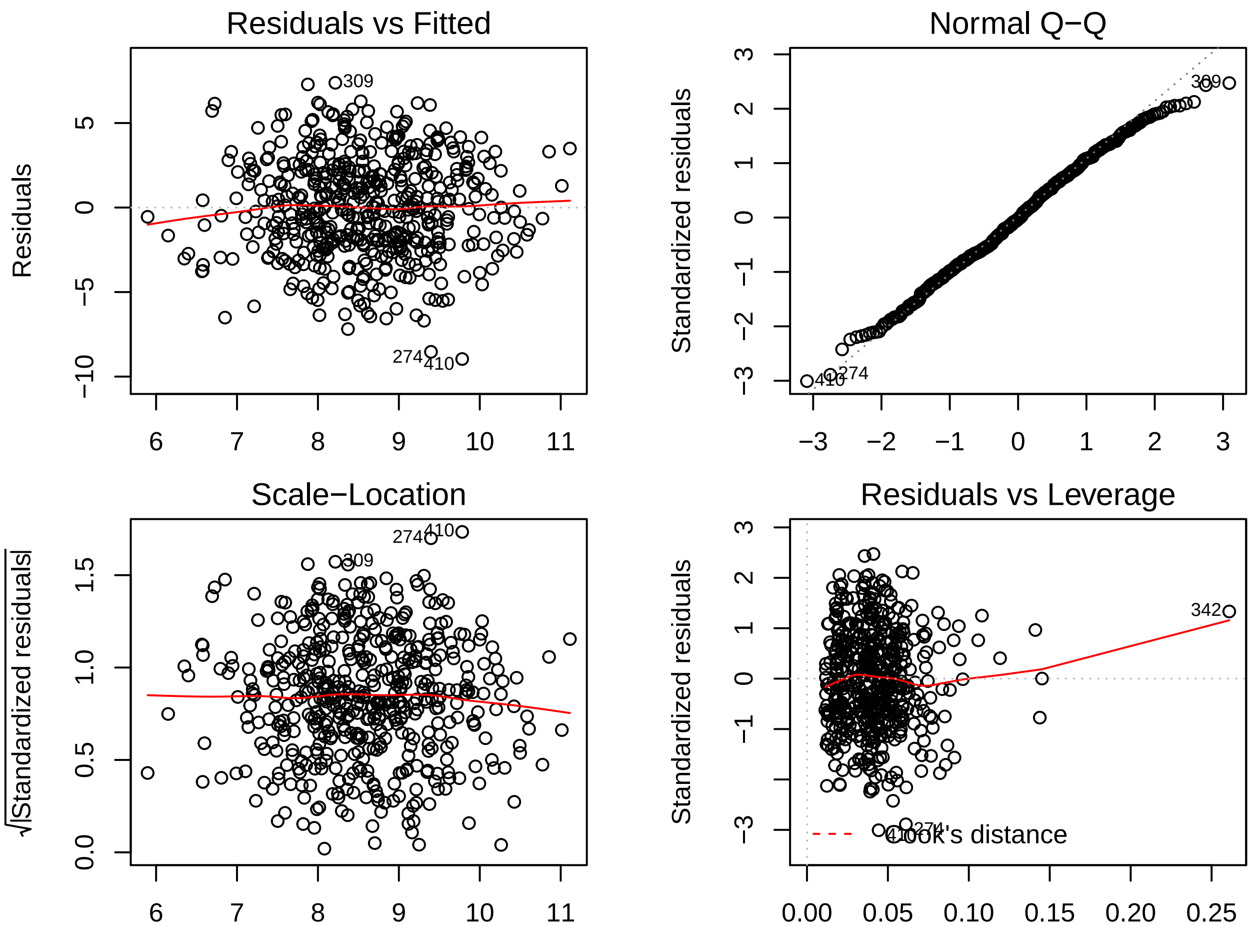
The Box-Cox transform selects a power transformation of ~ 0.741 for the response. We use 3/4, because it is close to the computed transform but is a more straight forward power transformation.

```r
fit.bc <- lm(dti^(3/4) ~ home_ownership + tax_liens +
                 inq_last_6mths + log(1+tot_coll_amt) + log(earliest_cr_line) +
                 emp_length + us_regions +
                 sqrt(total_acc), data = lc.train)
print(summary(fit.bc))
```

```
##
## Call:
## lm(formula = dti^(3/4) ~ home_ownership + tax_liens + inq_last_6mths +
##     log(1 + tot_coll_amt) + log(earliest_cr_line) + emp_length +
##     us_regions + sqrt(total_acc), data = lc.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9604 -2.0723 -0.0795  2.1969  7.3793
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            7.050155   1.155443   6.102 2.17e-09 ***
## home_ownershipOWN      0.068343   0.465725   0.147   0.8834
## home_ownershipRENT     0.112106   0.319588   0.351   0.7259
## tax_liens              0.852864   0.515146   1.656   0.0985 .
## inq_last_6mths        -0.045495   0.143750  -0.316   0.7518
## log(1 + tot_coll_amt) -0.007468   0.055938  -0.134   0.8939
## log(earliest_cr_line) -0.640610   0.348332  -1.839   0.0665 .
## emp_length1 year       1.715807   0.749878   2.288   0.0226 *
## emp_length10+ years    0.592625   0.549218   1.079   0.2811
## emp_length2 years      1.016694   0.663084   1.533   0.1259
## emp_length3 years      0.730580   0.676341   1.080   0.2806
## emp_length4 years     -0.700607   0.710581  -0.986   0.3246
## emp_length5 years      0.097772   0.715065   0.137   0.8913
## emp_length6 years      1.394993   0.889255   1.569   0.1174
## emp_length7 years      0.667120   0.780653   0.855   0.3932
## emp_length8 years      0.907357   0.712338   1.274   0.2034
## emp_length9 years      1.196452   0.819276   1.460   0.1448
## us_regionsNortheast   -0.775266   0.470731  -1.647   0.1002
## us_regionsSouth       -0.362581   0.400414  -0.906   0.3656
## us_regionsWest        -0.506633   0.439796  -1.152   0.2499
## sqrt(total_acc)        0.605189   0.142761   4.239 2.69e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 3.048 on 479 degrees of freedom
## Multiple R-squared:  0.07839,    Adjusted R-squared:  0.03991
## F-statistic: 2.037 on 20 and 479 DF,  p-value: 0.005251
```

```r
par(mfrow = c(2, 2), mar = c(2, 4.5, 2, 2))
plot(fit.bc)
```



There is no apparent pattern in the residuals versus fitted plot. The normal Q-Q plot shows very slight divergence from a linear fit in the upper quantiles.

```r
ncvTest(fit.bc)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.1689529, Df = 1, p = 0.68104
```

The ncvTest fails to reject homoskedasticity.

```r
shapiro.test(residuals(fit.bc))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit.bc)
## W = 0.99509, p-value = 0.1141
```

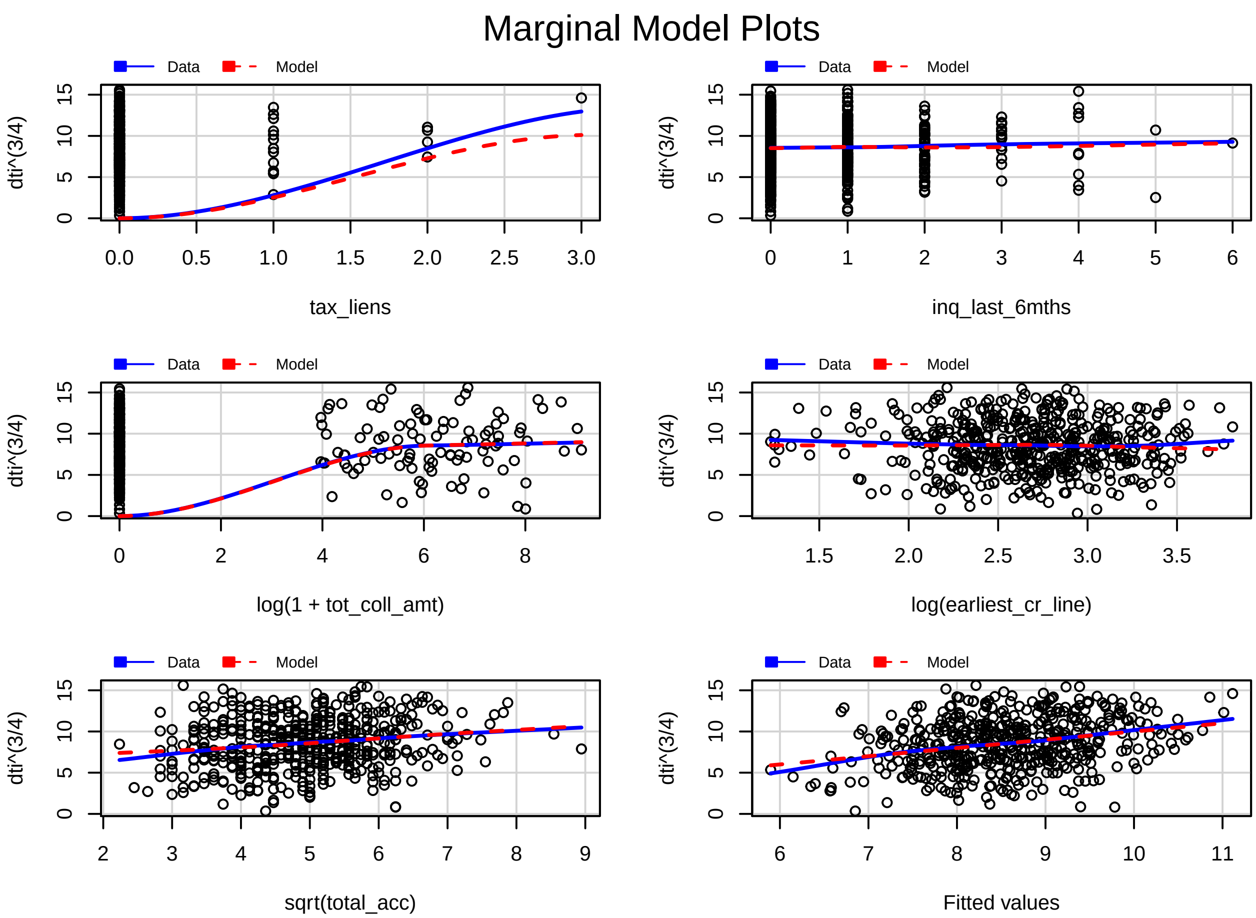
The Shapiro-Wilk test fails to reject the normality of residuals.

```
resettest(fit.bc)
```

```
##
##  RESET test
##
## data:  fit.bc
## RESET = 1.4903, df1 = 2, df2 = 477, p-value = 0.2263
```

The reset test fails to reject the hypothesis that the model doesn't need transformation.

```
mmps(fit.bc, terms = ~ tax_liens + inq_last_6mths +
                  log(1+tot_coll_amt) + log(earliest_cr_line) +
                  sqrt(total_acc))
```



Marginal model plots have a good fit for all predictors except tax_liens.

Stepwise BIC on the Manually Transformed Model

```
step.bic<- step(fit.bc, k = log(nrow(lc.train)))
```

```
## Start:  AIC=1223.58
```

```
## dti^(3/4) ~ home_ownership + tax_liens + inq_last_6mths + log(1 +
##     tot_coll_amt) + log(earliest_cr_line) + emp_length + us_regions +
##     sqrt(total_acc)
##
##                         Df Sum of Sq    RSS    AIC
## - emp_length            10   147.279 4597.7 1177.7
## - us_regions             3    26.626 4477.0 1207.9
## - home_ownership         2     1.171 4451.6 1211.3
## - log(1 + tot_coll_amt)  1     0.166 4450.6 1217.4
## - inq_last_6mths         1     0.931 4451.3 1217.5
## - tax_liens              1    25.466 4475.9 1220.2
## - log(earliest_cr_line)  1    31.424 4481.8 1220.9
## <none>                              4450.4 1223.6
## - sqrt(total_acc)        1   166.967 4617.4 1235.8
##
## Step:  AIC=1177.71
## dti^(3/4) ~ home_ownership + tax_liens + inq_last_6mths + log(1 +
##     tot_coll_amt) + log(earliest_cr_line) + us_regions + sqrt(total_acc)
##
##                         Df Sum of Sq    RSS    AIC
## - us_regions             3    24.674 4622.4 1161.7
## - home_ownership         2     0.150 4597.8 1165.3
## - inq_last_6mths         1     0.024 4597.7 1171.5
## - log(1 + tot_coll_amt)  1     0.131 4597.8 1171.5
## - tax_liens              1    23.225 4620.9 1174.0
## - log(earliest_cr_line)  1    40.205 4637.9 1175.8
## <none>                              4597.7 1177.7
## - sqrt(total_acc)        1   172.592 4770.3 1189.9
##
## Step:  AIC=1161.74
## dti^(3/4) ~ home_ownership + tax_liens + inq_last_6mths + log(1 +
##     tot_coll_amt) + log(earliest_cr_line) + sqrt(total_acc)
##
##                         Df Sum of Sq    RSS    AIC
## - home_ownership         2     0.167 4622.5 1149.3
## - log(1 + tot_coll_amt)  1     0.018 4622.4 1155.5
## - inq_last_6mths         1     0.031 4622.4 1155.5
## - tax_liens              1    18.919 4641.3 1157.6
## - log(earliest_cr_line)  1    43.295 4665.7 1160.2
## <none>                              4622.4 1161.7
## - sqrt(total_acc)        1   179.817 4802.2 1174.6
##
## Step:  AIC=1149.33
## dti^(3/4) ~ tax_liens + inq_last_6mths + log(1 + tot_coll_amt) +
##     log(earliest_cr_line) + sqrt(total_acc)
##
##                         Df Sum of Sq    RSS    AIC
## - log(1 + tot_coll_amt)  1     0.015 4622.5 1143.1
```

```
## - inq_last_6mths         1      0.043 4622.6 1143.1
## - tax_liens              1     19.167 4641.7 1145.2
## - log(earliest_cr_line)  1     44.359 4666.9 1147.9
## <none>                                 4622.5 1149.3
## - sqrt(total_acc)        1    182.588 4805.1 1162.5
##
## Step:  AIC=1143.12
## dti^(3/4) ~ tax_liens + inq_last_6mths + log(earliest_cr_line) +
##     sqrt(total_acc)
##
##                         Df Sum of Sq    RSS    AIC
## - inq_last_6mths         1      0.037 4622.6 1136.9
## - tax_liens              1     19.220 4641.8 1139.0
## - log(earliest_cr_line)  1     44.459 4667.0 1141.7
## <none>                                 4622.5 1143.1
## - sqrt(total_acc)        1    182.604 4805.1 1156.3
##
## Step:  AIC=1136.91
## dti^(3/4) ~ tax_liens + log(earliest_cr_line) + sqrt(total_acc)
##
##                         Df Sum of Sq    RSS    AIC
## - tax_liens              1     19.262 4641.8 1132.8
## - log(earliest_cr_line)  1     44.734 4667.3 1135.5
## <none>                                 4622.6 1136.9
## - sqrt(total_acc)        1    185.676 4808.3 1150.4
##
## Step:  AIC=1132.77
## dti^(3/4) ~ log(earliest_cr_line) + sqrt(total_acc)
##
##                         Df Sum of Sq    RSS    AIC
## - log(earliest_cr_line)  1     41.788 4683.6 1131.0
## <none>                                 4641.8 1132.8
## - sqrt(total_acc)        1    185.433 4827.3 1146.1
##
## Step:  AIC=1131.04
## dti^(3/4) ~ sqrt(total_acc)
##
##                   Df Sum of Sq    RSS    AIC
## <none>                          4683.6 1131.0
## - sqrt(total_acc)  1    145.31 4828.9 1140.1
```

```r
summary(step.bic)
```

```
##
## Call:
## lm(formula = dti^(3/4) ~ sqrt(total_acc), data = lc.train)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max 
## -8.4201 -2.1446  0.0358  2.1574  7.9214 
## 
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)        6.0631     0.6541   9.270  < 2e-16 ***
## sqrt(total_acc)    0.5091     0.1295   3.931 9.67e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 3.067 on 498 degrees of freedom
## Multiple R-squared:  0.03009,    Adjusted R-squared:  0.02814 
## F-statistic: 15.45 on 1 and 498 DF,  p-value: 9.666e-05
```

The output above indicate what the model selection process.

```r
BIC(fit.bc)
```

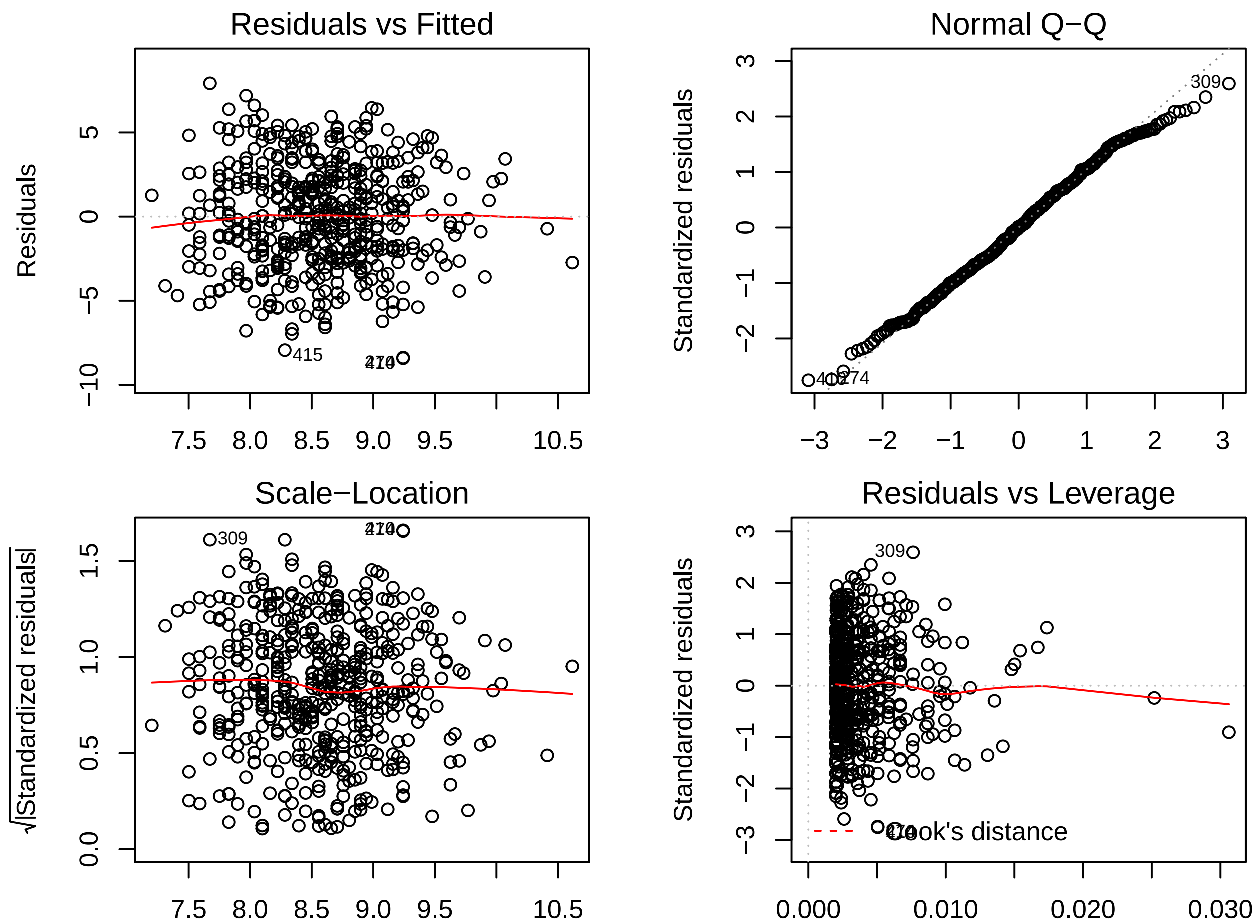
```
## [1] 2648.731
```

```r
BIC(step.bic)
```

```
## [1] 2556.193
```

The stepwise regression optimizes for smaller BIC. BIC strikes a balance between model complexity and explanatory power when comparing model candidates.

```r
par(mfrow = c(2, 2), mar = c(2, 4.5, 2, 2))
plot(step.bic)
```



Assumptions of modeling are not violated as residuals are randomly scattered. The normal QQ plot looks linear with slight deviations in upper and lower quantiles. The residuals vs leverage plot shows a few points outside the (-2,+2) range which indicates potential outliers that might be worth looking at. Particularly notable are point 309 and point 270.

Stepwise AIC on the Manually Transformed Model

Reduce the number of extraneous parameters using stepwise regression based on AIC.

```r
fit.transformed.aic <- step(fit.bc)
```

```
## Start:  AIC=1135.07
## dti^(3/4) ~ home_ownership + tax_liens + inq_last_6mths + log(1 +
##     tot_coll_amt) + log(earliest_cr_line) + emp_length + us_regions +
##     sqrt(total_acc)
##
##                         Df Sum of Sq    RSS    AIC
## - home_ownership         2     1.171 4451.6 1131.2
## - emp_length            10   147.279 4597.7 1131.3
## - us_regions             3    26.626 4477.0 1132.0
## - log(1 + tot_coll_amt)  1     0.166 4450.6 1133.1
## - inq_last_6mths         1     0.931 4451.3 1133.2
```

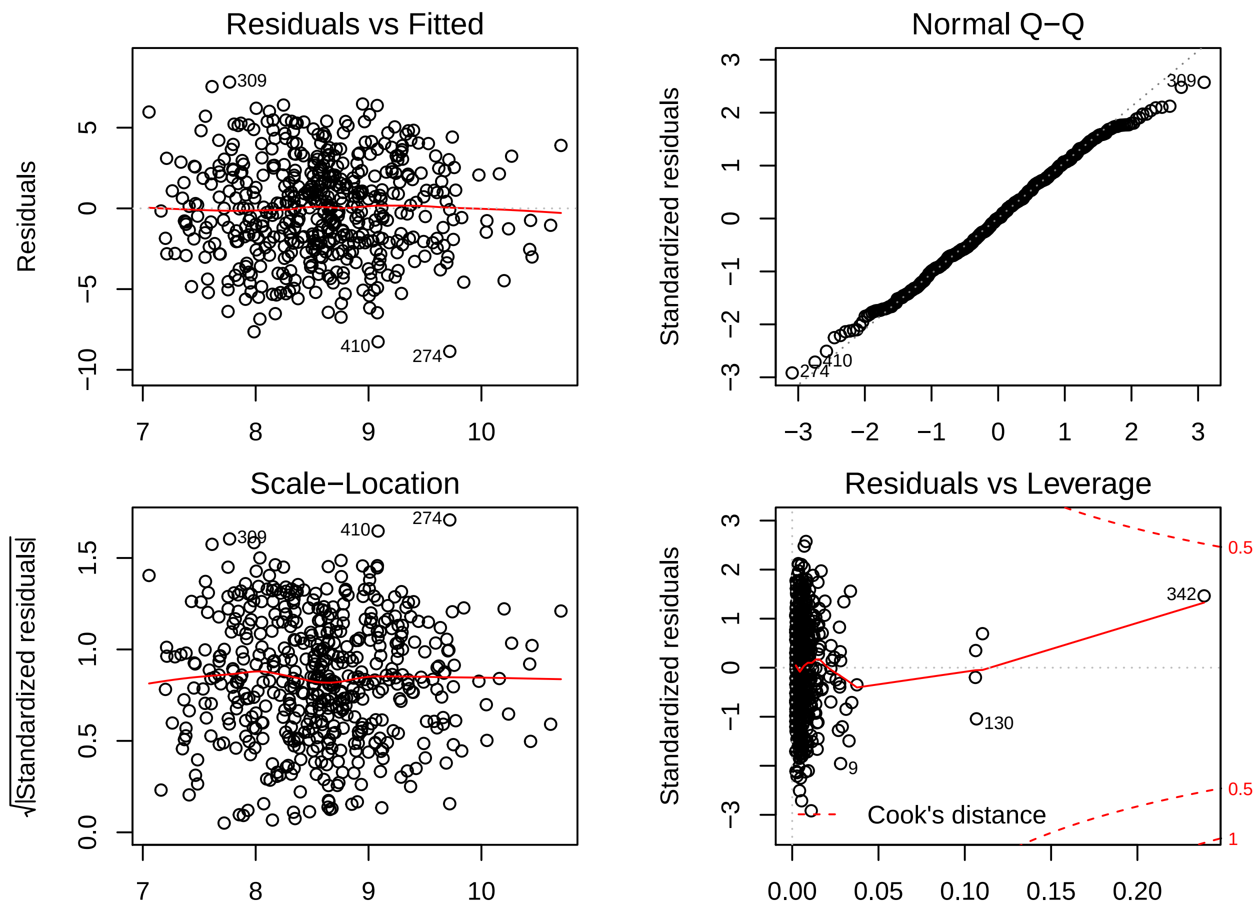
```
## <none>                              4450.4 1135.1
## - tax_liens              1    25.466 4475.9 1135.9
## - log(earliest_cr_line)  1    31.424 4481.8 1136.6
## - sqrt(total_acc)        1   166.967 4617.4 1151.5
##
## Step:  AIC=1131.2
## dti^(3/4) ~ tax_liens + inq_last_6mths + log(1 + tot_coll_amt) +
##     log(earliest_cr_line) + emp_length + us_regions + sqrt(total_acc)
##
##                         Df Sum of Sq    RSS    AIC
## - emp_length            10   146.259 4597.8 1127.4
## - us_regions             3    25.771 4477.3 1128.1
## - log(1 + tot_coll_amt)  1     0.153 4451.7 1129.2
## - inq_last_6mths         1     0.992 4452.6 1129.3
## <none>                              4451.6 1131.2
## - tax_liens              1    25.017 4476.6 1132.0
## - log(earliest_cr_line)  1    34.567 4486.1 1133.1
## - sqrt(total_acc)        1   166.010 4617.6 1147.5
##
## Step:  AIC=1127.37
## dti^(3/4) ~ tax_liens + inq_last_6mths + log(1 + tot_coll_amt) +
##     log(earliest_cr_line) + us_regions + sqrt(total_acc)
##
##                         Df Sum of Sq    RSS    AIC
## - us_regions             3    24.690 4622.5 1124.0
## - inq_last_6mths         1     0.023 4597.9 1125.4
## - log(1 + tot_coll_amt)  1     0.123 4598.0 1125.4
## <none>                              4597.8 1127.4
## - tax_liens              1    23.200 4621.0 1127.9
## - log(earliest_cr_line)  1    42.542 4640.4 1130.0
## - sqrt(total_acc)        1   173.611 4771.4 1143.9
##
## Step:  AIC=1124.04
## dti^(3/4) ~ tax_liens + inq_last_6mths + log(1 + tot_coll_amt) +
##     log(earliest_cr_line) + sqrt(total_acc)
##
##                         Df Sum of Sq    RSS    AIC
## - log(1 + tot_coll_amt)  1     0.015 4622.5 1122.0
## - inq_last_6mths         1     0.043 4622.6 1122.0
## <none>                              4622.5 1124.0
## - tax_liens              1    19.167 4641.7 1124.1
## - log(earliest_cr_line)  1    44.359 4666.9 1126.8
## - sqrt(total_acc)        1   182.588 4805.1 1141.4
##
## Step:  AIC=1122.05
## dti^(3/4) ~ tax_liens + inq_last_6mths + log(earliest_cr_line) +
##     sqrt(total_acc)
##
```

```
##                         Df Sum of Sq    RSS    AIC
## - inq_last_6mths         1     0.037 4622.6 1120.0
## <none>                               4622.5 1122.0
## - tax_liens              1    19.220 4641.8 1122.1
## - log(earliest_cr_line)  1    44.459 4667.0 1124.8
## - sqrt(total_acc)        1   182.604 4805.1 1139.4
##
## Step:  AIC=1120.05
## dti^(3/4) ~ tax_liens + log(earliest_cr_line) + sqrt(total_acc)
##
##                         Df Sum of Sq    RSS    AIC
## <none>                               4622.6 1120.0
## - tax_liens              1    19.262 4641.8 1120.1
## - log(earliest_cr_line)  1    44.734 4667.3 1122.9
## - sqrt(total_acc)        1   185.676 4808.3 1137.7
```

```r
print(summary(fit.transformed.aic))
```

```
##
## Call:
## lm(formula = dti^(3/4) ~ tax_liens + log(earliest_cr_line) +
##     sqrt(total_acc), data = lc.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8581 -2.1126  0.0168  2.2003  7.8253
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             7.4057     0.8989   8.238 1.57e-15 ***
## tax_liens               0.7242     0.5037   1.438   0.1512
## log(earliest_cr_line)  -0.7273     0.3320  -2.191   0.0289 *
## sqrt(total_acc)         0.6241     0.1398   4.464 9.99e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.053 on 496 degrees of freedom
## Multiple R-squared:  0.04273,    Adjusted R-squared:  0.03694
## F-statistic: 7.381 on 3 and 496 DF,  p-value: 7.588e-05
```

```r
par(mfrow = c(2, 2), mar = c(2, 4.5, 2, 2))
plot(fit.transformed.aic)
```



The resulting model has only three predictors. Both sets of categorical predictors have been removed from the model. The normal Q-Q plot has a good fit.

```
ncvTest(fit.transformed.aic)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 1.024092, Df = 1, p = 0.31155
```

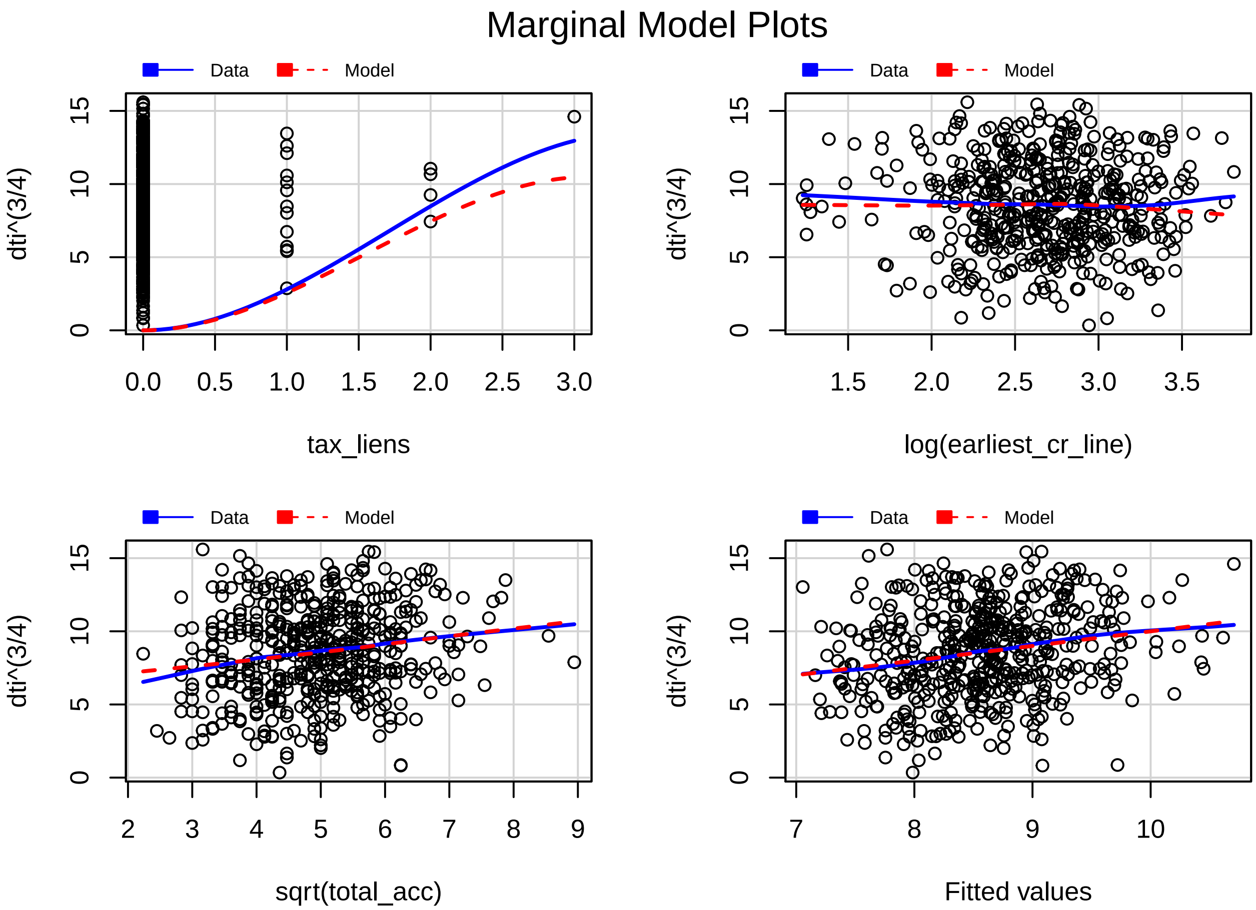
We fail to reject homoskedasticity.

```
shapiro.test(residuals(fit.transformed.aic))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fit.transformed.aic)
## W = 0.99485, p-value = 0.09298
```

The Shapiro-Wilk test fails to reject normality of residuals.

```
mmps(fit.transformed.aic, ~ tax_liens + log(earliest_cr_line) +
                              sqrt(total_acc))
```



Marginal model plots show a mismatch for `tax_liens`.

Multivariate Box-Cox Transformation of Predictors and Response to Normality Simultaneously

We use the multivariate Box-Cox transform to simultaneously transform the predictors and response to normality. In order to apply the Box-Cox transformation, all columns under consideration have to be positive. Some data points are zero in our data set. They are transformed by adding one. We exclude categorical predictors from the transformation.

```
powers.bc <- powerTransform(cbind(dti, (tax_liens + 1),
                                  (inq_last_6mths + 1),
                                  (tot_coll_amt + 1),
                                  (earliest_cr_line + 1),
                                  (total_acc + 1)) ~ 1,
                            data = lc.train)
powers.bc
```

```
## Estimated transformation parameters
##          dti
##   0.74103180 -33.80274515  -1.38836400  -0.78816824   0.09217768
##
```

```
##   0.21154806
```

We approximate the suggested transformation powers with values close to simple fractions: 3/4 for `dti`, -33 for `(tax_liens + 1)`, -4/3 for `(inq_last_6_mths + 1)`, -4/5 for `(tot_coll_amt + 1)`, 1/10 for `(earliest_cr_line + 1)` and 1/5 for `(total_acc + 1)`.

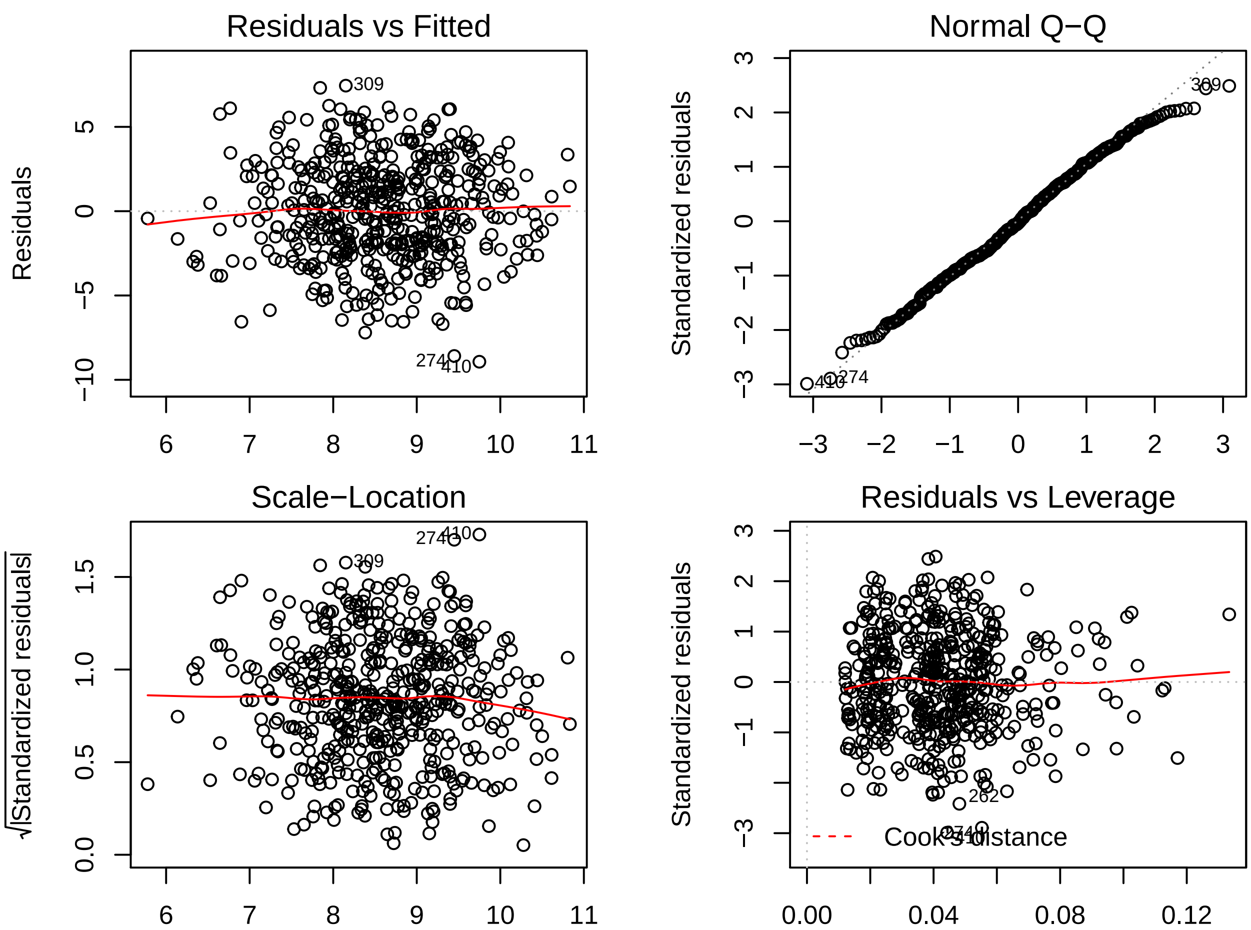
We then fit another model based on the coefficients from the Box-Cox transformation

```
fit.simultaneously.transformed <- lm(dti^(3/4) ~ I((tax_liens + 1)^-33) +
                               I((inq_last_6mths + 1)^(-4/3)) + emp_length + home_ownership +
                               I((tot_coll_amt + 1)^-(4/5)) + us_regions +
                               I((earliest_cr_line + 1)^(1/10)) +
                               I((total_acc + 1)^(1/5)), data = lc.train)
print(summary(fit.simultaneously.transformed))
```

```
##
## Call:
## lm(formula = dti^(3/4) ~ I((tax_liens + 1)^-33) + I((inq_last_6mths +
##     1)^(-4/3)) + emp_length + home_ownership + I((tot_coll_amt +
##     1)^-(4/5)) + us_regions + I((earliest_cr_line + 1)^(1/10)) +
##     I((total_acc + 1)^(1/5)), data = lc.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9275 -2.0470 -0.0992  2.1386  7.4434
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      7.743738   3.782078   2.047   0.0412 *
## I((tax_liens + 1)^-33)          -0.593715   0.752736  -0.789   0.4307
## I((inq_last_6mths + 1)^(-4/3))   0.110654   0.418233   0.265   0.7915
## emp_length1 year                 1.725267   0.750566   2.299   0.0220 *
## emp_length10+ years              0.585842   0.549447   1.066   0.2869
## emp_length2 years                1.001571   0.663396   1.510   0.1318
## emp_length3 years                0.713706   0.677510   1.053   0.2927
## emp_length4 years               -0.697944   0.710120  -0.983   0.3262
## emp_length5 years                0.137785   0.717053   0.192   0.8477
## emp_length6 years                1.404125   0.892024   1.574   0.1161
## emp_length7 years                0.662291   0.783328   0.845   0.3983
## emp_length8 years                0.880835   0.714675   1.232   0.2184
## emp_length9 years                1.183106   0.820595   1.442   0.1500
## home_ownershipOWN                0.063078   0.466537   0.135   0.8925
## home_ownershipRENT               0.107360   0.320185   0.335   0.7375
## I((tot_coll_amt + 1)^-(4/5))    -0.004717   0.360277  -0.013   0.9896
## us_regionsNortheast             -0.716540   0.469944  -1.525   0.1280
## us_regionsSouth                 -0.346422   0.401254  -0.863   0.3884
## us_regionsWest                  -0.499952   0.440816  -1.134   0.2573
## I((earliest_cr_line + 1)^(1/10)) -5.055135   2.875619  -1.758   0.0794 .
## I((total_acc + 1)^(1/5))         4.065989   0.960848   4.232 2.78e-05 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.054 on 479 degrees of freedom
## Multiple R-squared:  0.07465,    Adjusted R-squared:  0.03601
## F-statistic: 1.932 on 20 and 479 DF,  p-value: 0.009234
```

```r
par(mfrow = c(2, 2), mar = c(2, 4.5, 2, 2))
plot(fit.simultaneously.transformed)
```



The residuals vs fitted plot does not show any obvious pattern and has a horizontal trend line. The normal Q-Q plot shows a linear relationship with slight curvature in the upper and lower quantiles.

```r
ncvTest(fit.simultaneously.transformed)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.1819234, Df = 1, p = 0.66973
```

We fail to reject homoskedasticity of the model.

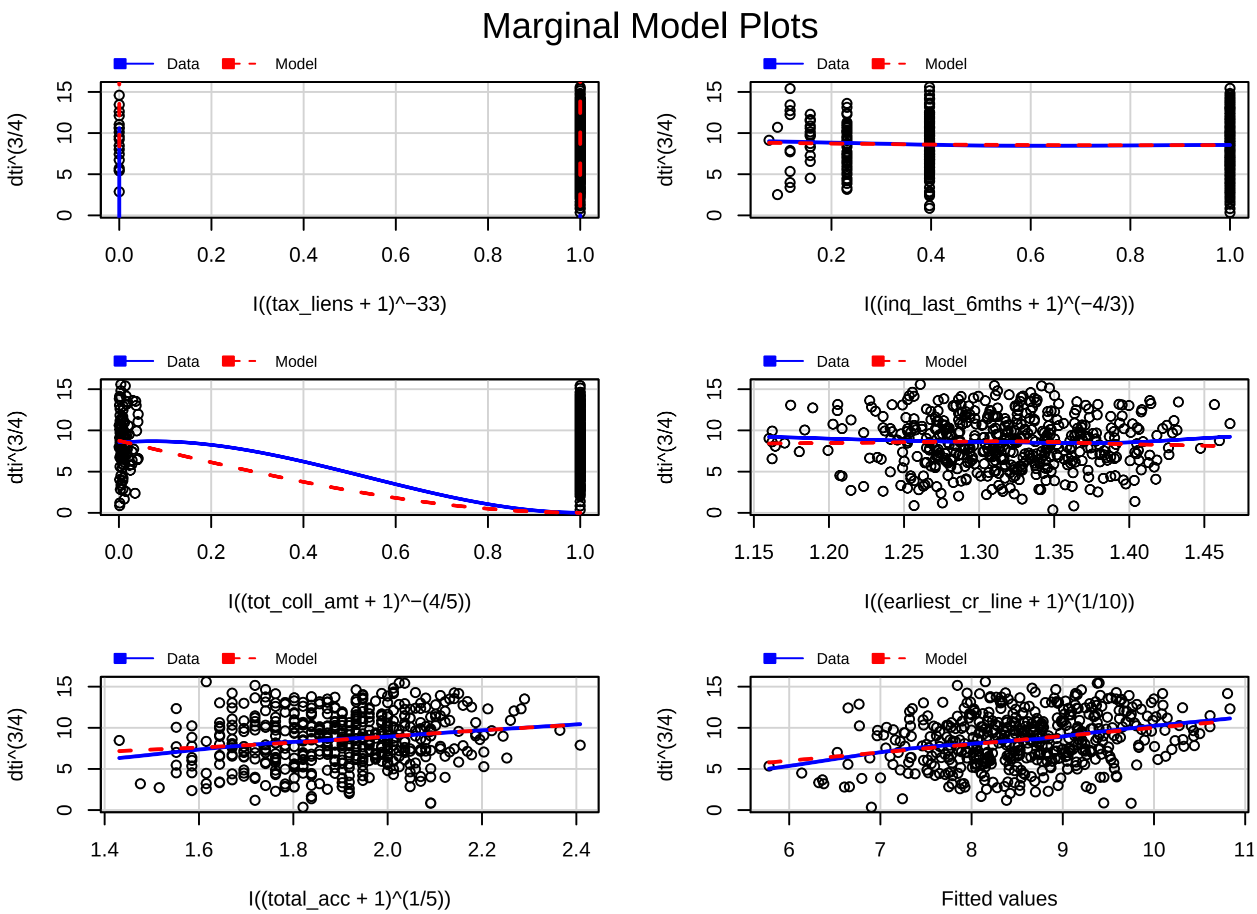
```r
shapiro.test(residuals(fit.simultaneously.transformed))
```

```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  residuals(fit.simultaneously.transformed)
## W = 0.99498, p-value = 0.1042
```

The Shapiro-Wilk test fails to reject normality of the residuals.

```
mmps(fit.simultaneously.transformed, ~ I((tax_liens + 1)^-33) +
                                       I((inq_last_6mths + 1)^(-4/3)) +
                                       I((tot_coll_amt + 1)^-(4/5)) +
                                       I((earliest_cr_line + 1)^(1/10)) +
                                       I((total_acc + 1)^(1/5)))
```



The marginal plots show a good fit for every predictor except `tot_coll_amt` and `tax_liens`.

BIC from All First Order Interactions

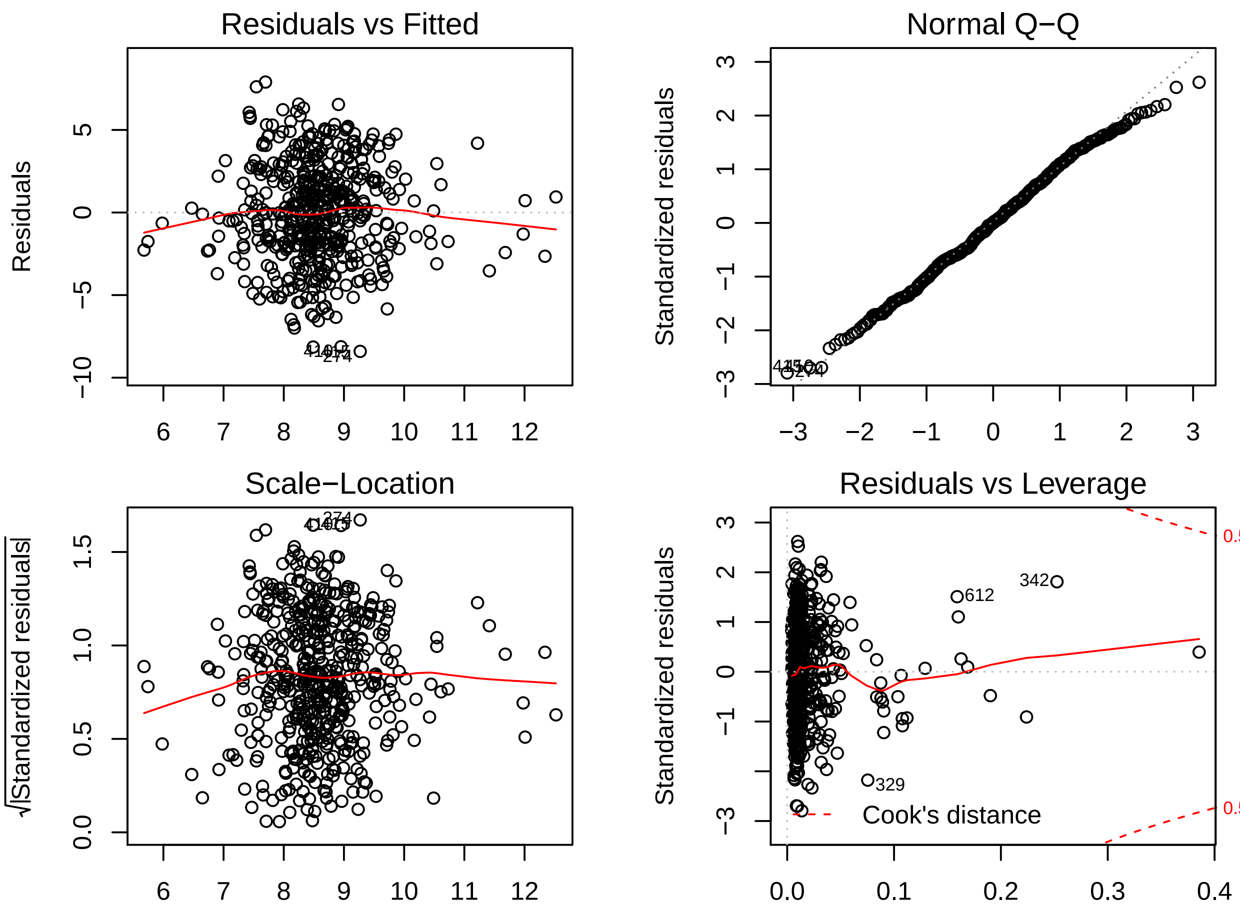
We use stepwise regression starting with all first-order interactions based on the Bayesian information criterion. We then use the transformation of the response selected by Box-Cox previously so that models have the same scale when doing model comparison.

```
initial.fit <- lm(dti^(3/4) ~ .*., data = lc.train)
final.bic.fit <- step(initial.fit, k = log(nrow(lc.train)), trace = FALSE)
```

```r
print(summary(final.bic.fit))
```

```
##
## Call:
## lm(formula = dti^(3/4) ~ home_ownership + tax_liens + earliest_cr_line +
##     total_acc + inq_last_6mths + home_ownership:inq_last_6mths +
##     tax_liens:earliest_cr_line, data = lc.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.4081 -2.0035  0.0122  2.1706  7.8971
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     7.67126    0.48802  15.719  < 2e-16 ***
## home_ownershipOWN              -0.59728    0.57074  -1.046  0.29585
## home_ownershipRENT              0.52969    0.36069   1.469  0.14260
## tax_liens                      -3.96427    1.94346  -2.040  0.04191 *
## earliest_cr_line               -0.03870    0.02142  -1.807  0.07134 .
## total_acc                       0.05379    0.01335   4.030 6.46e-05 ***
## inq_last_6mths                  0.18722    0.18162   1.031  0.30312
## home_ownershipOWN:inq_last_6mths  0.56496    0.40842   1.383  0.16721
## home_ownershipRENT:inq_last_6mths -0.87470    0.31173  -2.806  0.00522 **
## tax_liens:earliest_cr_line      0.26326    0.10527   2.501  0.01272 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.03 on 490 degrees of freedom
## Multiple R-squared:  0.06847,    Adjusted R-squared:  0.05136
## F-statistic: 4.002 on 9 and 490 DF,  p-value: 5.953e-05
```

```r
par(mfrow = c(2, 2), mar = c(2, 4.5, 2, 2))
plot(final.bic.fit)
```



The residuals vs fit shows randomly distributed points with a slight curved trend. The normal Q-Q plot shows a strong linear relationship.

```
ncvTest(final.bic.fit)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.6581199, Df = 1, p = 0.41722
```

The ncvTest fails to reject homoskedasticity of the model.

```
shapiro.test(residuals(final.bic.fit))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(final.bic.fit)
## W = 0.99556, p-value = 0.168
```

The Shapiro-Wilk test fails to reject normality of residuals.

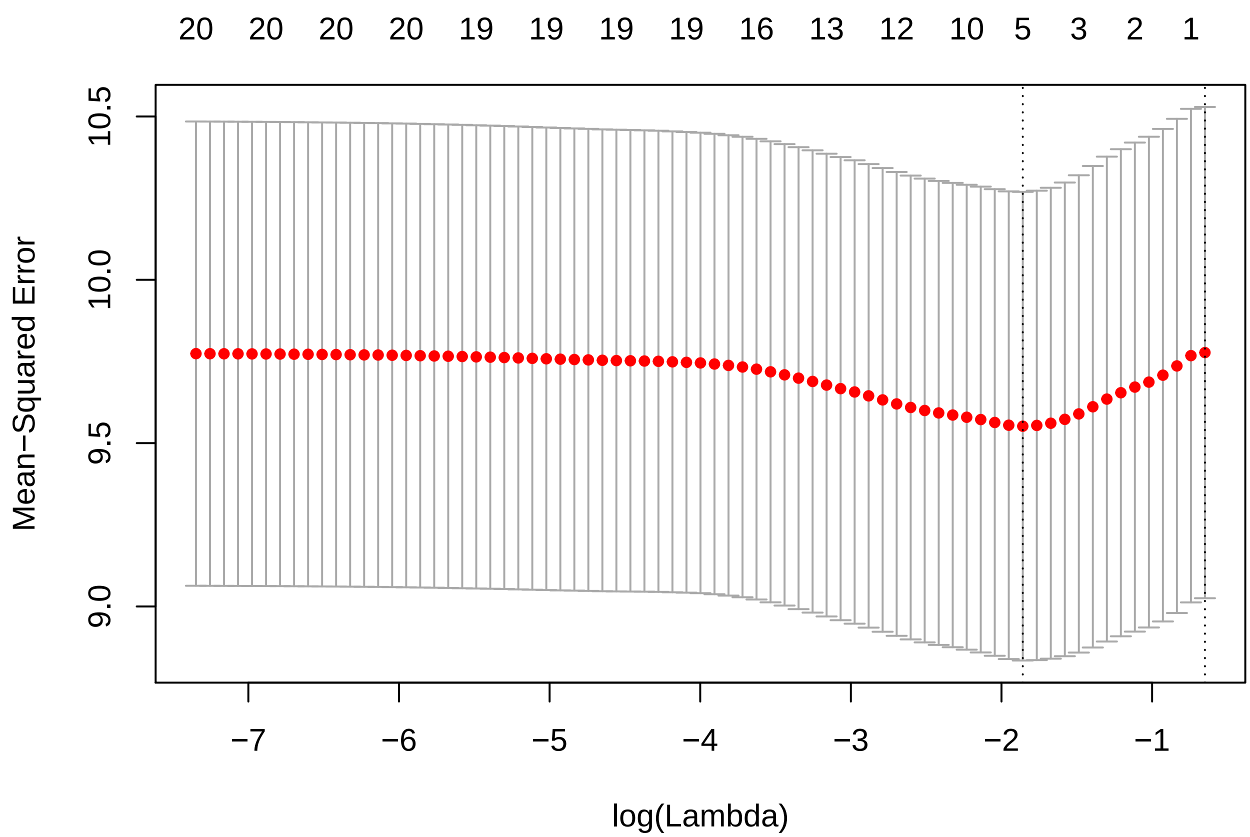
V. Lasso Regression

First we set up the data for glmnet. We then use the same transformation of the response that was selected by the Box-Cox transform to enable comparison of models with RMSE. Lasso stands for "least absolute shrinkage and selection operator". It performs both variable selection and regularization. Coefficients that are close to zero are dropped from the model.

```
library(glmnet)
X <- model.matrix(dti^(3/4)~., lc.train)[,-1]
Y <- lc.train$dti^(3/4)
```

We examine cross-validation error. The lowest point in the curve indicates the optimal log lambda value for the model.

```
cv <- cv.glmnet(X,Y,alpha=1)
plot(cv)
```




```
cv$lambda.min
```

```
## [1] 0.1558822
```

The optimal value of lambda is `cv$lambda.min`. We fit a model using the selected value of lambda and the prepared X and Y values.

```
model <- glmnet(X,Y,alpha=1,lambda=cv$lambda.min)
```

We use RMSE for predictions on a test data set to assess the predictive power of the model.

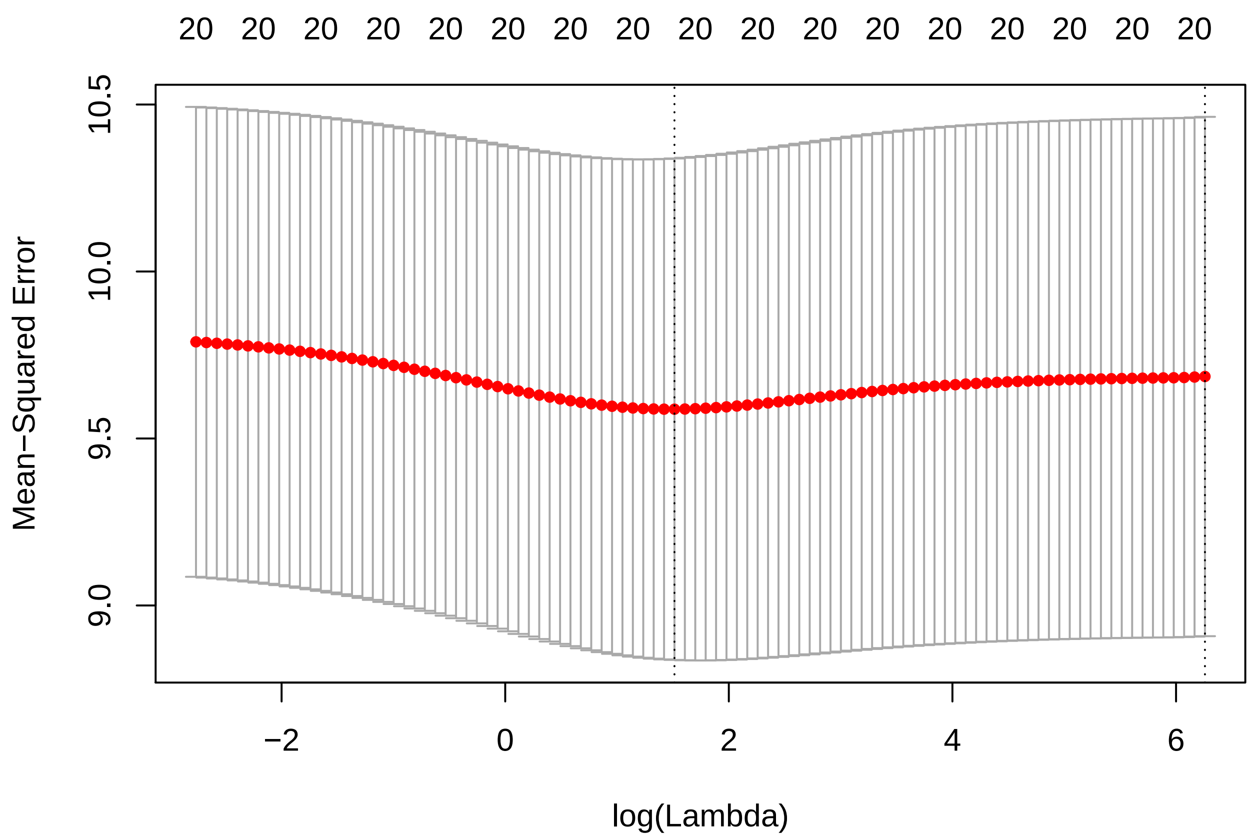
```
X.test <- model.matrix(dti^(3/4)~., lc.test)
predictions.lasso <- X.test%*%coef(model)
RMSE_lasso <- sqrt(mean((predictions.lasso-lc.test$dti^(3/4))^2))
RMSE_lasso
```

```
## [1] 2.915458
```

VI. Ridge Regression

We examine k-fold cross validation error in a ridge regression and establish the optimal value of lambda. Ridge regression is similar to Lasso in the sense that it is another form of regularization, however, coefficients cannot be eliminated from a ridge regression model whereas they can in Lasso regression.

```
cv <- cv.glmnet(X,Y,alpha=0)
plot(cv)
```




```
cv$lambda.min
```

```
## [1] 4.544049
```

The optimal value of lambda is `cv$lambda.min`. We fit a ridge regression model with the selected lambda value.

```
model <- glmnet(X,Y,alpha=0,lambda=cv$lambda.min)
```

Again, we assess the predictive power of the model using the RMSE of predictions on test data.

```
X.test <- model.matrix(dti^(3/4)~.,lc.test)
fits.ridge <- X.test%*%coef(model)
RMSE_ridge <- sqrt(mean((fits.ridge-lc.test$dti^(3/4))^2))
RMSE_ridge
```

```
## [1] 2.924822
```

VII. Neural Network Models

We scale the data to improve the fit of the neural network model. We also create dummy variables for the categorical inputs because the neural network library doesn't work directly with factor variables.

```
require(nnet)
normalize <- function(x) { (x - min(x)) / (max(x) - min(x))}
lc.normalized <- as.data.frame(lapply(lc.selected[,c('tax_liens','earliest_cr_line',
    'total_acc','inq_last_6mths','tot_coll_amt','dti')], normalize))
lc.normalized$ho_mortgage <- ifelse(lc.selected$home_ownership == 'MORTGAGE', 1, 0)
lc.normalized$ho_own <- ifelse(lc.selected$home_ownership == 'OWN', 1, 0)
lc.normalized$r_mw <- ifelse(lc.selected$us_regions == 'Midwest', 1, 0)
lc.normalized$r_s <- ifelse(lc.selected$us_regions == 'South', 1, 0)
lc.normalized$r_w <- ifelse(lc.selected$us_regions == 'West', 1, 0)
lc.train.normalized <- lc.normalized[train.rows,]
lc.test.normalized <- lc.normalized[test.rows,]
```

We fit three models with two, four and six hidden nodes respectively and then assess their performance on the test data.

```
set.seed(123487)
fit.nnet.2 = neuralnet::neuralnet(dti ~ tax_liens + earliest_cr_line + total_acc +
                                    inq_last_6mths + tot_coll_amt + ho_mortgage +
                                    ho_own + r_mw + r_s + r_w, data = lc.train.normalized,
                               linear.output = TRUE, hidden = c(2,1), threshold = 0.01)
fit.nnet.4 = neuralnet::neuralnet(dti ~ tax_liens + earliest_cr_line + total_acc +
                                      inq_last_6mths +
                                    tot_coll_amt + ho_mortgage + ho_own + r_mw +
                                    r_s + r_w, data = lc.train.normalized,
                                  linear.output = TRUE, hidden = c(4,1), threshold = 0.01)
fit.nnet.6 = neuralnet::neuralnet(dti ~ tax_liens + earliest_cr_line + total_acc +
                                    inq_last_6mths + tot_coll_amt +
                                    ho_mortgage + ho_own + r_mw + r_s +
                                    r_w, data = lc.train.normalized, linear.output = TRUE,
                                hidden = c(6,1), threshold = 0.01 )
```

To determine RMSE for neural network models with the same scale and location as the other models we apply the inverse of the normalization transformation to the predictions. We then apply the same power transformation as other models used on the response and compute RMSE.

```
normalize.inverse <- function(dti.prediction) {
    dti.prediction * (max(lc.test$dti) - min(lc.test$dti)) + min(lc.test$dti)
}
nnet.rmse <- function(predicted) {
    denormalized <- normalize.inverse(predicted)
    transformed <- denormalized^.75
    sqrt(mean((transformed - lc.test$dti^.75)^2))
}
```

We evaluate these models using RMSE on the test data set.

```
lc.test.pred.2 = predict(fit.nnet.2, newdata = lc.test.normalized, type = "response")
rmse.nnet.hidden.2 <- nnet.rmse(lc.test.pred.2)
lc.test.pred.4 = predict(fit.nnet.4, newdata = lc.test.normalized, type = "response")
rmse.nnet.hidden.4 <- nnet.rmse(lc.test.pred.4)
lc.test.pred.6 = predict(fit.nnet.6, newdata = lc.test.normalized, type = "response")
rmse.nnet.hidden.6 <- nnet.rmse(lc.test.pred.6)
print(c(rmse.nnet.hidden.2, rmse.nnet.hidden.4, rmse.nnet.hidden.6))
```

```
## [1] 2.934043 3.060384 3.242687
```

VIII. Model Comparison

We compare the models with a **transformed response** using model probability based on the Bayesian information criterion.

```
models <- list(final.bic.fit, fit.bc, fit.simultaneously.transformed,
               fit.transformed.aic, step.bic)
bic <- sapply(models, function(model) { extractAIC(model, k = log(nrow(lc.train)))[2] })
print(bic)
```

```
## [1] 1160.568 1223.578 1225.605 1136.909 1131.040
```

```
eBIC <- exp(-0.5 * (bic - min(bic)))
print(eBIC)
```

```
## [1] 3.872005e-07 8.043869e-21 2.919545e-21 5.315728e-02 1.000000e+00
```

```
probs <- eBIC / sum(eBIC)
round(probs, 5)
```

```
## [1] 0.00000 0.00000 0.00000 0.05047 0.94953
```

The best model based on Bayesian model probability is the stepwise regression using BIC starting from the manually transformed predictors and the Box-Cox transformed response.

We again compare models using RMSE on the test data.

```r
RMSE <- function(model) {
    prediction <- predict(model, newdata = lc.test)
    sqrt(mean((prediction-lc.test$dti^(3/4))^2))
}
models <- list(final.bic.fit, fit.bc, fit.simultaneously.transformed, step.bic,
               fit.transformed.aic)
ols.models.RMSE <- sapply(models, RMSE)
c(RMSE_lasso, RMSE_ridge, rmse.nnet.hidden.2,
  rmse.nnet.hidden.4, rmse.nnet.hidden.6, ols.models.RMSE)
```

```
##  [1] 2.915458 2.924822 2.934043 3.060384 3.242687 2.928206 2.891146
##  [8] 2.875387 2.916441 2.901913
```

The best predictive model based on RMSE for the test data is the model where the response and predictors were simultaneously transformed to normality with the multivariate Box-Cox transform.

IX. Expectations and Observed Outcomes

We assess expectations and observed effects in the model with predictors and response simultaneously transformed using the Box-Cox transformation, which had the best predictive performance. Of the predictors we included in this model very few had significant p-values at the 5% or 10% levels. These predictors were `employment_length1 year` at the 5% level, `earliest_cr_line` at the 10% level and `total_acc` at the 5% level. These are the only predictors for which this model provides evidence of an effect.

We expected that longer durations of employment would result in reduced DTI. The only significant level of this categorical variable was an employment length of one year, the was the second lowest duration. The model provides evidence that compared to longer employment durations this employment duration was related to an increase in DTI. This agrees with our intuition about the effect of employment length. Because of the variable transformations, the effect of this variable is difficult to quantify.

We expected that loan applicants with a longer duration between `earliest_cr_line` and `issue_d` have had time to stabilize their earnings and develop financial maturity which we expected to correlate with lower DTI. The transformed `earliest_cr_line` predictor is significant at the 10% level providing some evidence that it has an effect. The sign of the coefficient in the model is negative providing evidence that the longer the duration between an applicant's earliest credit line and their loan issue date, the lower their DTI. Again this agrees with our expectations.

We expected that `total_acc` being the number of credit lines a borrower has open would be correlated with increased DTI. This predictor is highly significant in the model. The complicated transformation of both this variable and the response make it difficult to quantify the effect of this predictor. However, the positive coefficient along with the strictly increasing transformation provide evidence that an increase in this predictor does result in an increase in DTI.

For each predictor which had evidence of an effect in the model, our intuition agreed with the sign of the effect.

The predictive performance of this model hints at the possibility that other predictors, although not statistically significant do have a relationship with DTI and it would be interesting to do further analysis to attempt to establish whether this is the case.

X. Conclusion

We applied a transformation to the response because without this the Shapiro-Wilk test indicated that the assumption of normality of residuals was violated.

Of the models investigated here, the best predictive model was the model based on a multivariate Box-Cox transformation of the predictors and response simultaneously. While this model has good predictive power, the complex power transformations of the predictors make it difficult to interpret the effect of the predictors.

References

1. https://www.lendingclub.com/info/download-data.action
2. https://help.lendingclub.com/hc/en-us/articles/216127307-Data-Dictionaries
3. https://www.globalbankingandfinance.com/the-future-is-here-ai-and-machine-learning-in-financial-servic
4. Sheather, Simon J. *A Modern Approach to Regression with R.* Springer, 2010.
5. Kutner, Michael H., et al. *Applied Linear Statistical Models.* Irwin, 1996.
6. Caffo, Brian. *Statistical inference for data science.* Leanpub, 2016.
7. Pardoe, Iain. *Applied Regression Modeling: A Business Approach.* Wiley, 2006.
8. Frees, Edward W. *Regression Modeling with Acturial and Financial Applications.* Cambridge, 2010.
9. Kiva.org: Crowd-Sourced Microfinance & Cooperation in Group Lending