Contents


```
# summarize data
summary(loan)
```

```
##    loan_amnt      funded_amnt    funded_amnt_inv        term
##  Min.   :  500   Min.   :  500   Min.   :    0   36_months:28301
##  1st Qu.: 5500   1st Qu.: 5500   1st Qu.: 5000   60_months:10670
##  Median :10000   Median : 9800   Median : 9000
##  Mean   :11267   Mean   :10991   Mean   :10567
##  3rd Qu.:15000   3rd Qu.:15000   3rd Qu.:14500
##  Max.   :35000   Max.   :35000   Max.   :35000
##
##     int_rate       installment     grade       sub_grade
##  Min.   :0.0542   Min.   :  16.1   A: 9937   B3     : 2886
##  1st Qu.:0.0925   1st Qu.: 167.4   B:11810   A4     : 2856
##  Median :0.1186   Median : 280.9   C: 7889   A5     : 2706
##  Mean   :0.1205   Mean   : 325.1   D: 5185   B5     : 2665
##  3rd Qu.:0.1461   3rd Qu.: 431.4   E: 2805   B4     : 2465
##  Max.   :0.2459   Max.   :1305.2   F: 1033   C1     : 2086
##                                    G:  312   (Other):23307
##              emp_title         emp_length     home_ownership
##                   : 2406   10+ years: 8770   MORTGAGE:17427
##  US Army          :  131   < 1 year : 4404   OTHER   :   96
##  Bank of America  :  108   2 years  : 4299   OWN     : 2992
##  IBM              :   66   3 years  : 4026   RENT    :18456
##  AT&T             :   60   4 years  : 3385
##  Kaiser Permanente:   56   5 years  : 3239
##  (Other)          :36144   (Other)  :10848
##    annual_inc           verification_status     issue_d
##  Min.   :   4000   Not Verified   :16165    Dec 2011: 2266
##  1st Qu.:  40800   Source Verified: 9992    Nov 2011: 2226
##  Median :  59386   Verified       :12814    Oct 2011: 2113
##  Mean   :  69038                            Sep 2011: 2061
##  3rd Qu.:  82500                            Aug 2011: 1933
##  Max.   :6000000                            Jul 2011: 1866
##                                             (Other) :26506
##       loan_status                   purpose         zip_code
##  Charged Off: 5468   debt_consolidation:18345   100xx  :  570
```

```
##  Fully Paid :33503   credit_card       : 5034   945xx  :  540
##                      other             : 3875   112xx  :  505
##                      home_improvement  : 2929   606xx  :  498
##                      major_purchase    : 2158   070xx  :  458
##                      small_business    : 1768   900xx  :  441
##                      (Other)           : 4862   (Other):35959
##    addr_state          dti          delinq_2yrs     earliest_cr_line
##  CA     : 6995   Min.   : 0.00   Min.   : 0.000   Nov 1998:  366
##  NY     : 3721   1st Qu.: 8.25   1st Qu.: 0.000   Oct 1999:  363
##  FL     : 2816   Median :13.46   Median : 0.000   Dec 1998:  343
##  TX     : 2676   Mean   :13.37   Mean   : 0.145   Oct 2000:  339
##  NJ     : 1814   3rd Qu.:18.63   3rd Qu.: 0.000   Dec 1997:  321
##  IL     : 1510   Max.   :29.99   Max.   :11.000   Nov 1999:  316
##  (Other):19439                                     (Other) :36923
##  inq_last_6mths     open_acc        pub_rec         revol_bal
##  Min.   :0.000   Min.   : 2.0   Min.   :0.000   Min.   :     0
##  1st Qu.:0.000   1st Qu.: 6.0   1st Qu.:0.000   1st Qu.:  3746
##  Median :1.000   Median : 9.0   Median :0.000   Median :  8897
##  Mean   :0.866   Mean   : 9.3   Mean   :0.056   Mean   : 13414
##  3rd Qu.:1.000   3rd Qu.:12.0   3rd Qu.:0.000   3rd Qu.: 17095
##  Max.   :8.000   Max.   :44.0   Max.   :4.000   Max.   :149588
##
##    revol_util      total_acc     total_pymnt    total_pymnt_inv
##  Min.   :0.000   Min.   : 2.0   Min.   :   34   Min.   :    0
##  1st Qu.:0.256   1st Qu.:14.0   1st Qu.: 5605   1st Qu.: 5338
##  Median :0.495   Median :21.0   Median : 9991   Median : 9520
##  Mean   :0.490   Mean   :22.2   Mean   :12297   Mean   :11836
##  3rd Qu.:0.725   3rd Qu.:29.0   3rd Qu.:16689   3rd Qu.:16096
##  Max.   :0.999   Max.   :90.0   Max.   :58886   Max.   :58564
##
##  total_rec_prncp total_rec_int   total_rec_late_fee   recoveries
##  Min.   :    0   Min.   :    6   Min.   :  0.00     Min.   :    0
##  1st Qu.: 4727   1st Qu.:  668   1st Qu.:  0.00     1st Qu.:    0
##  Median : 8000   Median : 1360   Median :  0.00     Median :    0
##  Mean   : 9904   Mean   : 2296   Mean   :  1.35     Mean   :   96
##  3rd Qu.:14000   3rd Qu.: 2875   3rd Qu.:  0.00     3rd Qu.:    0
##  Max.   :35000   Max.   :23886   Max.   :180.20     Max.   :29623
##
##  collection_recovery_fee   last_pymnt_d   last_pymnt_amnt
##  Min.   :   0            Mar 2013: 1024   Min.   :    0
##  1st Qu.:   0            Dec 2014:  944   1st Qu.:  221
##  Median :   0            May 2013:  907   Median :  558
##  Mean   :  12            Feb 2013:  868   Mean   : 2707
##  3rd Qu.:   0            Apr 2013:  851   3rd Qu.: 3349
##  Max.   :7002            Mar 2012:  843   Max.   :36115
##                          (Other) :33534
##  last_credit_pull_d pub_rec_bankruptcies
##  Apr 2017: 9312     Min.   :0.0000
##  Oct 2016: 4238     1st Qu.:0.0000
##  Feb 2017: 1101     Median :0.0000
##  Mar 2017:  870     Mean   :0.0432
##  Jan 2017:  658     3rd Qu.:0.0000
##  Dec 2016:  636     Max.   :2.0000
##  (Other) :22156
```

```r
# look at types
str(loan)
```

```
## Classes 'data.table' and 'data.frame':   38971 obs. of  38 variables:
##  $ loan_amnt              : int  5000 2500 2400 10000 3000 5000 7000 3000 5600 5375 ...
##  $ funded_amnt            : int  5000 2500 2400 10000 3000 5000 7000 3000 5600 5375 ...
##  $ funded_amnt_inv        : num  4975 2500 2400 10000 3000 ...
##  $ term                   : Factor w/ 2 levels "36_months","60_months": 1 2 1 1 2 1 2 1 2 2 ...
##  $ int_rate               : num  0.106 0.153 0.16 0.135 0.127 ...
##  $ installment            : num  162.9 59.8 84.3 339.3 67.8 ...
##  $ grade                  : Factor w/ 7 levels "A","B","C","D",..: 2 3 3 3 2 1 3 5 6 2 ...
##  $ sub_grade              : Factor w/ 35 levels "A1","A2","A3",..: 7 14 15 11 10 4 15 21 27 10 ...
##  $ emp_title              : Factor w/ 28303 levels "","  old palm inc",..: 1 18662 1 329 23262 23645
##  $ emp_length             : Factor w/ 12 levels "1 year","10+ years",..: 2 11 2 2 1 4 9 10 5 11 ...
##  $ home_ownership         : Factor w/ 4 levels "MORTGAGE","OTHER",..: 4 4 4 4 4 4 4 4 3 4 ...
##  $ annual_inc             : num  24000 30000 12252 49200 80000 ...
##  $ verification_status    : Factor w/ 3 levels "Not Verified",..: 3 2 1 2 2 2 1 2 2 3 ...
##  $ issue_d                : Factor w/ 52 levels "Apr 2008","Apr 2009",..: 14 14 14 14 14 14 14 14 14
##  $ loan_status            : Factor w/ 2 levels "Charged Off",..: 2 1 2 2 2 2 2 2 1 1 ...
##  $ purpose                : Factor w/ 14 levels "car","credit_card",..: 2 1 12 10 10 14 3 1 12 10 ..
##  $ zip_code               : Factor w/ 810 levels "007xx","010xx",..: 701 277 493 737 786 695 248 722
##  $ addr_state             : Factor w/ 49 levels "AK","AL","AR",..: 4 11 15 5 36 4 27 5 5 42 ...
##  $ dti                    : num  27.65 1 8.72 20 17.94 ...
##  $ delinq_2yrs            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ earliest_cr_line       : Factor w/ 526 levels "Apr 1964","Apr 1966",..: 192 34 426 161 203 429 254
##  $ inq_last_6mths         : int  1 5 2 1 0 3 1 2 2 0 ...
##  $ open_acc               : int  3 3 2 10 15 9 7 4 11 2 ...
##  $ pub_rec                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ revol_bal              : int  13648 1687 2956 5598 27783 7963 17726 8221 5210 9279 ...
##  $ revol_util             : num  0.837 0.094 0.985 0.21 0.539 0.283 0.856 0.875 0.326 0.365 ...
##  $ total_acc              : int  9 4 10 37 38 12 11 4 13 3 ...
##  $ total_pymnt            : num  5863 1015 3006 12232 4067 ...
##  $ total_pymnt_inv        : num  5834 1015 3006 12232 4067 ...
##  $ total_rec_prncp        : num  5000 456 2400 10000 3000 ...
##  $ total_rec_int          : num  863 435 606 2215 1067 ...
##  $ total_rec_late_fee     : num  0 0 0 17 0 ...
##  $ recoveries             : num  0 123 0 0 0 ...
##  $ collection_recovery_fee: num  0 1.11 0 0 0 0 0 0 2.09 2.52 ...
##  $ last_pymnt_d           : Factor w/ 107 levels "Apr 2009","Apr 2010",..: 43 5 61 43 45 43 80 43 4 8
##  $ last_pymnt_amnt        : num  171.6 119.7 649.9 357.5 67.3 ...
##  $ last_credit_pull_d     : Factor w/ 108 levels "Apr 2009","Apr 2010",..: 9 99 9 8 46 37 108 26 99 1
##  $ pub_rec_bankruptcies   : int  0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Data Cleaning

```r
# convert term into int
loan$term = as.numeric(gsub("([0-9]+)_.*", "\\1", loan$term))

# convert loan status to binary
loan$default <- ifelse(loan$loan_status=="Charged Off", 1, 0)
```

```
# convert date columns to date
loan$issue_d = paste("1", loan$issue_d, sep=" ")
loan$issue_d = as.Date(loan$issue_d, '%d %B %Y')

loan$earliest_cr_line = paste("1", loan$earliest_cr_line, sep=" ")
loan$earliest_cr_line = as.Date(loan$earliest_cr_line, '%d %B %Y')

loan$last_pymnt_d = paste("1", loan$last_pymnt_d, sep=" ")
loan$last_pymnt_d = as.Date(loan$last_pymnt_d, '%d %B %Y')

loan$last_credit_pull_d = paste("1", loan$last_credit_pull_d, sep=" ")
loan$last_credit_pull_d = as.Date(loan$last_credit_pull_d, '%d %B %Y')
```

```
# create data to illustrate map
map_data <- loan %>%
  group_by(addr_state, default) %>%
  tally() %>%
  group_by(addr_state) %>%
  mutate(pct = n / sum(n)) %>%
  select(addr_state, default, pct) %>%
  filter(default == 1)

# write.csv(map_data, file = "map_data.csv")
```

Hypotheses that I had

Hypothesis: Some loans were repaid early.

```
# function to add months to date
add.months= function(date,n) seq(date, by = paste (n, "months"), length = 2)[2]
# add.months(test_date,2)
```

```
# find out the date before loan expires
data_length = dim(loan)[1]
final_date = list()

i = 1
while(i < data_length + 1) {
    final_date[[i]] = add.months(loan$issue_d[i], loan$term[i])
    i = i + 1
}

library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
loan$final_date = as.Date(as.numeric(final_date))
```

```r
# find out if people paid early or late
difference = loan$final_date - loan$last_pymnt_d
loan$difference_d = difference

sum(difference > 0)
```

```
## [1] 25238
```

```r
sum(difference < 0)
```

```
## [1] 6664
```

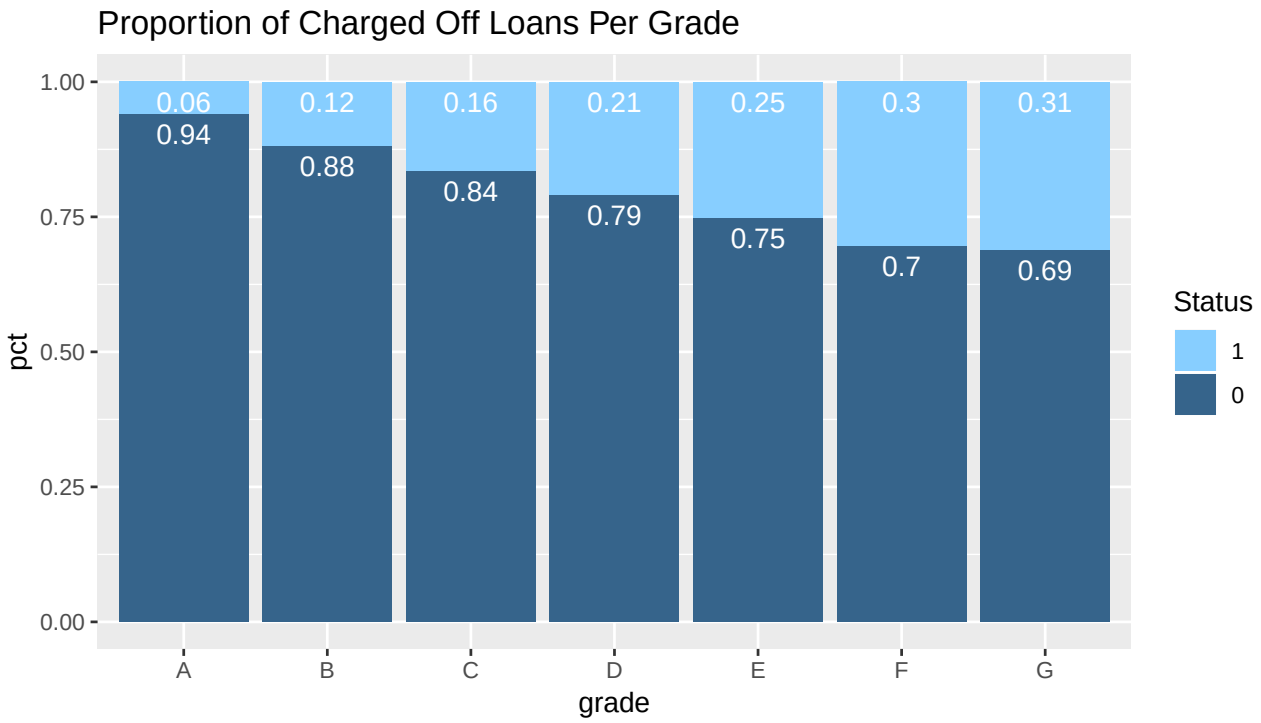
```r
sum(difference == 0)
```

```
## [1] 7069
```

From here, it's clear that most people paid off their loans before the term expired, meaning that they'll pay less interest.

Hypothesis: There is a strong relationship between loan default and grade group

```r
# proportion of defaults in each grade group
proportion <- loan %>%
        group_by(grade, default) %>%
        tally() %>%
        group_by(grade) %>%
        mutate(pct = n / sum(n)) %>%
        mutate(label_y = cumsum(pct))

ggplot(proportion, aes(grade, pct, fill = factor(default, levels = c(1,0)))) +
        geom_bar(stat = "identity") +
        geom_text(aes(label = round(pct, 2), y = label_y), vjust = 1.5, color = "white") +
        scale_fill_manual(values = c("#87CEFF", "#36648B")) +
        labs(fill = "Status") +
        ggtitle('Proportion of Charged Off Loans Per Grade')
```




```
# ggsave('proportion.png')
```

There appears to be a very strong relationship between default and grade.

```
model.grade = glm(default ~ grade, data=loan, family = 'binomial')
Anova(model.grade)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: default
##       LR Chisq Df          Pr(>Chisq)
## grade     1414  6 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

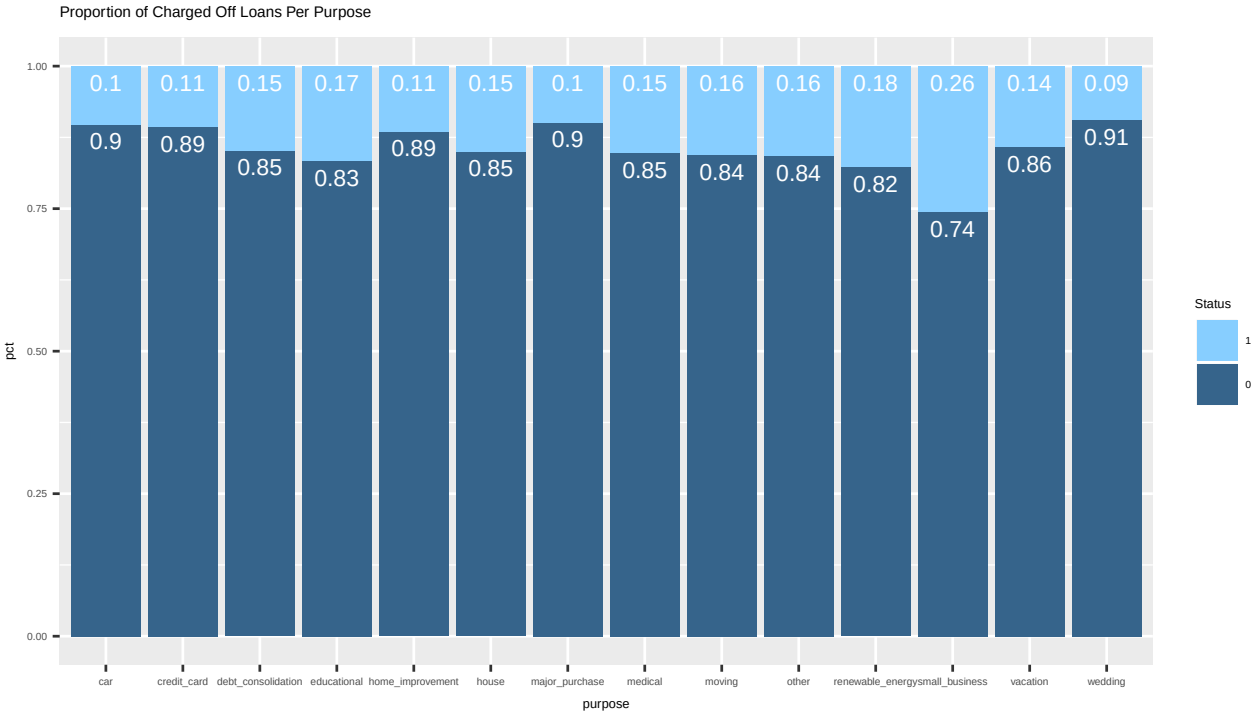
Hypothesis: There is a significant difference in default rates for each purpose

```
# are purpose and default significantly related?
model.purpose = glm(default ~ purpose, data=loan, family = 'binomial')
Anova(model.purpose)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: default
##         LR Chisq Df          Pr(>Chisq)
## purpose      327 13 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Grade appears to be significant.

```
# proportion of defaults for each purpose
proportion <- loan %>%
        group_by(purpose, default) %>%
        tally() %>%
        group_by(purpose) %>%
        mutate(pct = n / sum(n)) %>%
        mutate(label_y = cumsum(pct))

ggplot(proportion, aes(purpose, pct, fill = factor(default, levels = c(1,0)))) +
        geom_bar(stat = "identity") +
        geom_text(aes(label = round(pct, 2), y = label_y), vjust = 1.5, color = "white", size=3) +
        scale_fill_manual(values = c("#87CEFF", "#36648B")) +
        labs(fill = "Status") +
        ggtitle('Proportion of Charged Off Loans Per Purpose') +
        theme(text = element_text(size=5))
```




```
#ggsave('proportion_purpose.png')
```

The purpose of the loan appears to be significant.

Analyses

Goal 1: Predict which loans will default

```
# create dataframe for this analysis
loan_default = loan %>% select(-emp_title, -issue_d, -loan_status, -funded_amnt, -funded_amnt_inv, -tota

# train-test split
set.seed(471)
data_length = dim(loan_default)[1]
```

```r
propor = sample(1:data_length, 0.7*data_length)

# set up data
x = loan_default %>% select(-default)
y = loan_default %>% select(default)

x_train_default = x[propor,]
x_test_default = x[-propor,]

y_train_default = y[propor,]
y_test_default = y[-propor,]

loan_default_train = loan_default[propor,]
loan_default_test = loan_default[-propor,]
```

```r
str(x_train_default)
```

```
## Classes 'data.table' and 'data.frame':   27279 obs. of  18 variables:
##  $ loan_amnt           : int  7500 14000 5125 12000 4500 12000 30000 6000 10000 18000 ...
##  $ term                : num  36 60 36 36 60 36 36 36 36 60 ...
##  $ int_rate            : num  0.0991 0.1677 0.1186 0.1099 0.1372 ...
##  $ installment         : num  242 346 170 393 104 ...
##  $ grade               : Factor w/ 7 levels "A","B","C","D",..: 2 4 2 2 3 1 2 3 1 4 ...
##  $ home_ownership      : Factor w/ 4 levels "MORTGAGE","OTHER",..: 4 1 4 4 3 3 1 4 4 4 ...
##  $ annual_inc          : num  31200 147000 60000 78500 36000 ...
##  $ verification_status : Factor w/ 3 levels "Not Verified",..: 1 3 1 1 1 1 3 2 1 3 ...
##  $ purpose             : Factor w/ 14 levels "car","credit_card",..: 2 12 2 2 7 2 5 6 3 3 ...
##  $ dti                 : num  12.3 12.9 10.2 14.1 13.7 ...
##  $ delinq_2yrs         : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ inq_last_6mths      : int  0 1 0 1 2 1 3 2 1 0 ...
##  $ open_acc            : int  14 14 6 5 8 21 10 9 19 11 ...
##  $ pub_rec             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ revol_bal           : int  7790 4980 12674 0 4880 9913 17122 8032 13464 23719 ...
##  $ revol_util          : num  0.573 0.474 0.812 0 0.626 0.131 0.54 0.574 0.222 0.945 ...
##  $ total_acc           : int  18 34 17 11 10 40 37 37 49 37 ...
##  $ pub_rec_bankruptcies: int  0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```r
str(y_train_default)
```

```
## Classes 'data.table' and 'data.frame':   27279 obs. of  1 variable:
##  $ default: num  0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Model 1: Logistic Regression using variables from my hypotheses

```r
fit.logit.1 = glm(default ~ grade + purpose + inq_last_6mths, data=loan_default_train, family='binomial
summary(fit.logit.1)
```

```
##
## Call:
## glm(formula = default ~ grade + purpose + inq_last_6mths, family = "binomial",
##     data = loan_default_train)
##
## Deviance Residuals:
```

```
##    Min      1Q  Median      3Q     Max
## -1.313  -0.589  -0.484  -0.341   2.572
##
## Coefficients:
##                           Estimate Std. Error z value             Pr(>|z|)
## (Intercept)                -2.9698     0.1126  -26.38 < 0.0000000000000002
## gradeB                      0.7323     0.0612   11.97 < 0.0000000000000002
## gradeC                      1.0386     0.0633   16.42 < 0.0000000000000002
## gradeD                      1.3227     0.0661   20.02 < 0.0000000000000002
## gradeE                      1.6211     0.0730   22.20 < 0.0000000000000002
## gradeF                      1.8648     0.0969   19.24 < 0.0000000000000002
## gradeG                      1.8558     0.1600   11.60 < 0.0000000000000002
## purposecredit_card         -0.0943     0.1172   -0.81               0.4207
## purposedebt_consolidation   0.1530     0.1066    1.44               0.1513
## purposeeducational          0.5826     0.2038    2.86               0.0042
## purposehome_improvement    -0.0099     0.1248   -0.08               0.9368
## purposehouse                0.0892     0.2090    0.43               0.6694
## purposemajor_purchase      -0.0774     0.1354   -0.57               0.5675
## purposemedical              0.3364     0.1647    2.04               0.0411
## purposemoving               0.3306     0.1742    1.90               0.0578
## purposeother                0.3029     0.1165    2.60               0.0093
## purposerenewable_energy     0.6597     0.3208    2.06               0.0398
## purposesmall_business       0.7351     0.1236    5.95     0.00000000272797
## purposevacation             0.2124     0.2162    0.98               0.3258
## purposewedding             -0.3000     0.1737   -1.73               0.0842
## inq_last_6mths              0.1159     0.0161    7.21     0.00000000000057
##
## (Intercept)               ***
## gradeB                    ***
## gradeC                    ***
## gradeD                    ***
## gradeE                    ***
## gradeF                    ***
## gradeG                    ***
## purposecredit_card
## purposedebt_consolidation
## purposeeducational        **
## purposehome_improvement
## purposehouse
## purposemajor_purchase
## purposemedical            *
## purposemoving             .
## purposeother              **
## purposerenewable_energy   *
## purposesmall_business     ***
## purposevacation
## purposewedding            .
## inq_last_6mths            ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22112  on 27278  degrees of freedom
```

```
## Residual deviance: 20942  on 27258  degrees of freedom
## AIC: 20984
##
## Number of Fisher Scoring iterations: 5
```

```
#fit1.pred = ifelse(predict(fit.logit.5, x_test_default, type='response') >= 0.5, "1", "0")
#cm2 = table(fit2.pred, unlist(y_test_default)) # confusion matrix:
#error2 = (cm2[1,2]+cm2[2,1])/length(fit2.pred)
```

```
Anova(fit.logit.1)
```

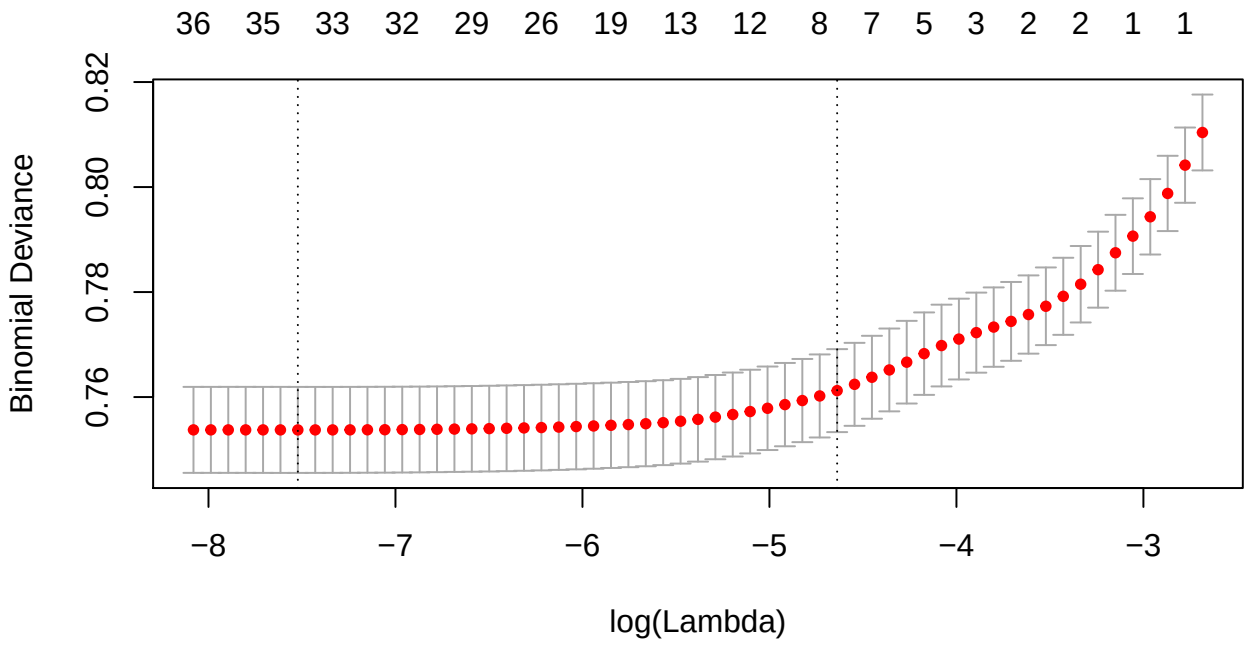
```
## Analysis of Deviance Table (Type II tests)
##
## Response: default
##                LR Chisq Df           Pr(>Chisq)
## grade               842  6 < 0.0000000000000002 ***
## purpose             135 13 < 0.0000000000000002 ***
## inq_last_6mths       50  1      0.0000000000012 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# rmse
pred1 = predict(fit.logit.1, x_test_default, type='response')
error1 = rmse(unlist(y_test_default), pred1)
error1
```

```
## [1] 0.3399
```

Model 2: Logistic Lasso

```
# prepare design matrix
X = model.matrix(default~., loan_default_train)[,-1]
Y = as.numeric(unlist(loan_default_train$default))

fit1.cv <- cv.glmnet(X, Y, alpha=1, family="binomial", nfolds = 10, type.measure = "deviance")
plot(fit1.cv)
```




```
# 1se coefficients
coef.1se = coef(fit1.cv, s="lambda.1se")
coef.1se = coef.1se[which(coef.1se !=0),]
coef.1se
```

```
##           (Intercept)                  term             int_rate
##          -3.789871670           0.016067318          10.191968963
##            annual_inc purposesmall_business        inq_last_6mths
##          -0.000002024           0.387987346           0.055792056
##               pub_rec            revol_util  pub_rec_bankruptcies
##           0.094895132           0.073518822           0.013097886
```

```
# min coefficients
coef.min = coef(fit1.cv, s="lambda.min")
coef.min = coef.min[which(coef.min !=0), ]
as.matrix(coef.min)
```

```
##                                              [,1]
## (Intercept)                          -4.343395349
## loan_amnt                             0.000002615
## term                                  0.022995239
## int_rate                             10.448882663
## gradeB                                0.109855494
## gradeC                                0.055512087
## gradeD                                0.027040949
## gradeG                               -0.106830786
## home_ownershipOTHER                   0.413187448
## home_ownershipOWN                     0.074886208
## home_ownershipRENT                    0.026335225
## annual_inc                           -0.000006240
## verification_statusSource Verified   -0.033491773
## verification_statusVerified           0.014487366
## purposecredit_card                   -0.169014396
## purposedebt_consolidation             0.034668053
## purposeeducational                    0.593052361
```

```
## purposemajor_purchase              -0.083206592
## purposemedical                      0.243390818
## purposemoving                       0.272358449
## purposeother                        0.239137859
## purposerenewable_energy             0.537785493
## purposesmall_business               0.752147354
## purposevacation                     0.112841469
## purposewedding                     -0.262764472
## dti                                 0.003205316
## delinq_2yrs                         0.009135269
## inq_last_6mths                      0.134533085
## open_acc                            0.006749439
## pub_rec                             0.239049757
## revol_bal                           0.000002718
## revol_util                          0.383876940
## total_acc                          -0.003451715
## pub_rec_bankruptcies                0.154309187
```

```
# variables using min
beta.min <- rownames(as.matrix(coef.min))
beta.min
```

```
##  [1] "(Intercept)"
##  [2] "loan_amnt"
##  [3] "term"
##  [4] "int_rate"
##  [5] "gradeB"
##  [6] "gradeC"
##  [7] "gradeD"
##  [8] "gradeG"
##  [9] "home_ownershipOTHER"
## [10] "home_ownershipOWN"
## [11] "home_ownershipRENT"
## [12] "annual_inc"
## [13] "verification_statusSource Verified"
## [14] "verification_statusVerified"
## [15] "purposecredit_card"
## [16] "purposedebt_consolidation"
## [17] "purposeeducational"
## [18] "purposemajor_purchase"
## [19] "purposemedical"
## [20] "purposemoving"
## [21] "purposeother"
## [22] "purposerenewable_energy"
## [23] "purposesmall_business"
## [24] "purposevacation"
## [25] "purposewedding"
## [26] "dti"
## [27] "delinq_2yrs"
## [28] "inq_last_6mths"
## [29] "open_acc"
## [30] "pub_rec"
## [31] "revol_bal"
## [32] "revol_util"
## [33] "total_acc"
```

```
## [34] "pub_rec_bankruptcies"
```

```
# logistic regression using variables from min
fit.logit.2 <- glm(default ~ loan_amnt + term + int_rate + grade + home_ownership + annual_inc + verifi
summary(fit.logit.2)
```

```
##
## Call:
## glm(formula = default ~ loan_amnt + term + int_rate + grade +
##     home_ownership + annual_inc + verification_status + purpose +
##     dti + delinq_2yrs + inq_last_6mths + open_acc + pub_rec +
##     revol_bal + revol_util + total_acc + pub_rec_bankruptcies,
##     family = "binomial", data = loan_default_train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.399  -0.586  -0.455  -0.323   4.300
##
## Coefficients:
##                                      Estimate   Std. Error z value
## (Intercept)                       -4.557742447  0.191043151  -23.86
## loan_amnt                          0.000003595  0.000003202    1.12
## term                               0.023891786  0.001909289   12.51
## int_rate                          10.094450801  1.804690429    5.59
## gradeB                             0.149384953  0.088606386    1.69
## gradeC                             0.095082277  0.125972250    0.75
## gradeD                             0.069296694  0.162287344    0.43
## gradeE                             0.025132445  0.195727318    0.13
## gradeF                             0.028588733  0.237948888    0.12
## gradeG                            -0.104107233  0.293017110   -0.36
## home_ownershipOTHER                0.498473301  0.328546404    1.52
## home_ownershipOWN                  0.099394025  0.071892225    1.38
## home_ownershipRENT                 0.039978657  0.042465440    0.94
## annual_inc                        -0.000006670  0.000000669   -9.96
## verification_statusSource Verified -0.043421646  0.047366617   -0.92
## verification_statusVerified         0.015159432  0.047552582    0.32
## purposecredit_card                 -0.019768929  0.120951778   -0.16
## purposedebt_consolidation           0.203521011  0.110185152    1.85
## purposeeducational                  0.799642681  0.205609193    3.89
## purposehome_improvement             0.191801752  0.127574804    1.50
## purposehouse                        0.214111879  0.212331878    1.01
## purposemajor_purchase               0.063167921  0.137177604    0.46
## purposemedical                      0.444508097  0.166849104    2.66
## purposemoving                       0.481453173  0.177076657    2.72
## purposeother                        0.422176958  0.118489974    3.56
## purposerenewable_energy             0.790159384  0.326582807    2.42
## purposesmall_business               0.939968297  0.126844331    7.41
## purposevacation                     0.341661260  0.218896599    1.56
## purposewedding                     -0.137470542  0.176043875   -0.78
## dti                                 0.002748215  0.003230923    0.85
## delinq_2yrs                         0.024717397  0.035984951    0.69
## inq_last_6mths                      0.140525586  0.016801342    8.36
## open_acc                            0.010687385  0.005867799    1.82
## pub_rec                             0.251059489  0.117447159    2.14
## revol_bal                           0.000003430  0.000001507    2.28
```

```
## revol_util                           0.406717748  0.085970821    4.73
## total_acc                           -0.004898112  0.002441891   -2.01
## pub_rec_bankruptcies                 0.161275960  0.137356383    1.17
##                                                        Pr(>|z|)
## (Intercept)                        < 0.0000000000000002 ***
## loan_amnt                                          0.26143
## term                               < 0.0000000000000002 ***
## int_rate                               0.00000002225977 ***
## gradeB                                             0.09181 .
## gradeC                                             0.45038
## gradeD                                             0.66938
## gradeE                                             0.89783
## gradeF                                             0.90437
## gradeG                                             0.72237
## home_ownershipOTHER                                0.12921
## home_ownershipOWN                                  0.16681
## home_ownershipRENT                                 0.34648
## annual_inc                         < 0.0000000000000002 ***
## verification_statusSource Verified                 0.35929
## verification_statusVerified                        0.74988
## purposecredit_card                                 0.87017
## purposedebt_consolidation                          0.06474 .
## purposeeducational                                 0.00010 ***
## purposehome_improvement                            0.13272
## purposehouse                                       0.31327
## purposemajor_purchase                              0.64517
## purposemedical                                     0.00772 **
## purposemoving                                      0.00655 **
## purposeother                                       0.00037 ***
## purposerenewable_energy                            0.01554 *
## purposesmall_business                  0.00000000000013 ***
## purposevacation                                    0.11856
## purposewedding                                     0.43487
## dti                                                0.39499
## delinq_2yrs                                        0.49216
## inq_last_6mths                     < 0.0000000000000002 ***
## open_acc                                           0.06855 .
## pub_rec                                            0.03255 *
## revol_bal                                          0.02281 *
## revol_util                             0.00000223547236 ***
## total_acc                                          0.04487 *
## pub_rec_bankruptcies                               0.24034
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22112  on 27278  degrees of freedom
## Residual deviance: 20473  on 27241  degrees of freedom
## AIC: 20549
##
## Number of Fisher Scoring iterations: 5
```

```r
# check which variables are significant
Anova(fit.logit.2)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: default
##                       LR Chisq Df           Pr(>Chisq)
## loan_amnt                1.3  1                0.262
## term                   154.7  1 < 0.0000000000000002 ***
## int_rate                31.5  1           0.00000002 ***
## grade                    9.0  6                0.171
## home_ownership           4.0  3                0.258
## annual_inc             120.7  1 < 0.0000000000000002 ***
## verification_status      1.5  2                0.464
## purpose                161.0 13 < 0.0000000000000002 ***
## dti                      0.7  1                0.395
## delinq_2yrs              0.5  1                0.494
## inq_last_6mths          67.9  1 < 0.0000000000000002 ***
## open_acc                 3.3  1                0.069 .
## pub_rec                  4.2  1                0.040 *
## revol_bal                5.0  1                0.025 *
## revol_util              22.6  1           0.00000204 ***
## total_acc                4.1  1                0.044 *
## pub_rec_bankruptcies     1.4  1                0.235
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# rmse
pred2 = predict(fit.logit.2, x_test_default, type='response')
error2 = rmse(unlist(y_test_default), pred2)
error2
```

```
## [1] 0.3373
```

```r
# model without non-significant variables
fit.logit.3 <- glm(default ~ term + int_rate + annual_inc + purpose + inq_last_6mths + pub_rec + revol_
summary(fit.logit.3)
```

```
##
## Call:
## glm(formula = default ~ term + int_rate + annual_inc + purpose +
##     inq_last_6mths + pub_rec + revol_bal + revol_util + total_acc,
##     family = "binomial", data = loan_default_train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.428  -0.579  -0.451  -0.337   4.345
##
## Coefficients:
##                               Estimate   Std. Error z value
## (Intercept)               -4.362873504  0.136167564  -32.04
## term                       0.023855894  0.001800634   13.25
## int_rate                  10.187835850  0.640515311   15.91
## annual_inc                -0.000006841  0.000000608  -11.25
## purposecredit_card         0.018363125  0.119433263    0.15
## purposedebt_consolidation  0.241027951  0.108340566    2.22
```

```
## purposeeducational         0.822893481  0.205209882    4.01
## purposehome_improvement    0.198273373  0.126409626    1.57
## purposehouse               0.231720237  0.211595640    1.10
## purposemajor_purchase      0.063285118  0.136823220    0.46
## purposemedical             0.460976903  0.166482455    2.77
## purposemoving              0.493002691  0.176451575    2.79
## purposeother               0.438221235  0.118107463    3.71
## purposerenewable_energy    0.805689130  0.326112796    2.47
## purposesmall_business      0.951849608  0.125499022    7.58
## purposevacation            0.335980772  0.218544121    1.54
## purposewedding            -0.116057692  0.175489240   -0.66
## inq_last_6mths             0.138792041  0.016604869    8.36
## pub_rec                    0.363868395  0.064230192    5.67
## revol_bal                  0.000004392  0.000001422    3.09
## revol_util                 0.375189642  0.079601753    4.71
## total_acc                 -0.001658104  0.001822034   -0.91
##                                      Pr(>|z|)
## (Intercept)               < 0.0000000000000002 ***
## term                      < 0.0000000000000002 ***
## int_rate                  < 0.0000000000000002 ***
## annual_inc                < 0.0000000000000002 ***
## purposecredit_card                    0.87781
## purposedebt_consolidation             0.02610 *
## purposeeducational           0.000060716426741 ***
## purposehome_improvement               0.11676
## purposehouse                          0.27347
## purposemajor_purchase                 0.64370
## purposemedical                        0.00562 **
## purposemoving                         0.00521 **
## purposeother                          0.00021 ***
## purposerenewable_energy               0.01349 *
## purposesmall_business        0.000000000000033 ***
## purposevacation                       0.12421
## purposewedding                        0.50840
## inq_last_6mths            < 0.0000000000000002 ***
## pub_rec                      0.000000014696636 ***
## revol_bal                             0.00201 **
## revol_util                   0.000002436963480 ***
## total_acc                             0.36281
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22112  on 27278  degrees of freedom
## Residual deviance: 20496  on 27257  degrees of freedom
## AIC: 20540
##
## Number of Fisher Scoring iterations: 5
```

```
Anova(fit.logit.3)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: default
```

```
##                LR Chisq Df           Pr(>Chisq)
## term              173.3  1 < 0.0000000000000002 ***
## int_rate          253.9  1 < 0.0000000000000002 ***
## annual_inc        155.6  1 < 0.0000000000000002 ***
## purpose           157.2 13 < 0.0000000000000002 ***
## inq_last_6mths     67.7  1 < 0.0000000000000002 ***
## pub_rec            30.1  1          0.000000042 ***
## revol_bal           9.2  1               0.0024 **
## revol_util         22.3  1          0.000002303 ***
## total_acc           0.8  1               0.3621
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# rmse
pred3 = predict(fit.logit.3, x_test_default, type='response')
error3 = rmse(unlist(y_test_default), pred3)
error3
```

```
## [1] 0.3373
```

```
# model without non-significant variables
fit.logit.4 <- glm(default ~ term + int_rate + annual_inc + purpose + inq_last_6mths + pub_rec + revol_
summary(fit.logit.4)
```

```
##
## Call:
## glm(formula = default ~ term + int_rate + annual_inc + purpose +
##     inq_last_6mths + pub_rec + revol_bal + revol_util, family = "binomial",
##     data = loan_default_train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.423  -0.579  -0.451  -0.337   4.385
##
## Coefficients:
##                               Estimate   Std. Error z value
## (Intercept)               -4.383541480  0.134290125  -32.64
## term                       0.023691807  0.001791258   13.23
## int_rate                  10.228890547  0.638709366   16.01
## annual_inc                -0.000006988  0.000000588  -11.88
## purposecredit_card         0.013883589  0.119312514    0.12
## purposedebt_consolidation  0.236058406  0.108184119    2.18
## purposeeducational         0.822906907  0.205181480    4.01
## purposehome_improvement    0.195479507  0.126351689    1.55
## purposehouse               0.233124502  0.211543893    1.10
## purposemajor_purchase      0.064175496  0.136802339    0.47
## purposemedical             0.458694409  0.166447184    2.76
## purposemoving              0.493357254  0.176421215    2.80
## purposeother               0.437363000  0.118087882    3.70
## purposerenewable_energy    0.800005700  0.325920636    2.45
## purposesmall_business      0.951692730  0.125486032    7.58
## purposevacation            0.334096385  0.218504169    1.53
## purposewedding            -0.114597158  0.175473310   -0.65
## inq_last_6mths             0.136861204  0.016469871    8.31
## pub_rec                    0.363064886  0.064237242    5.65
## revol_bal                  0.000004112  0.000001393    2.95
```

```
## revol_util                 0.384555132  0.078967715    4.87
##                                            Pr(>|z|)
## (Intercept)               < 0.0000000000000002 ***
## term                      < 0.0000000000000002 ***
## int_rate                  < 0.0000000000000002 ***
## annual_inc                < 0.0000000000000002 ***
## purposecredit_card                     0.90736
## purposedebt_consolidation              0.02911 *
## purposeeducational           0.000060557061668 ***
## purposehome_improvement                0.12184
## purposehouse                           0.27046
## purposemajor_purchase                  0.63899
## purposemedical                         0.00585 **
## purposemoving                          0.00517 **
## purposeother                           0.00021 ***
## purposerenewable_energy                0.01410 *
## purposesmall_business        0.000000000000033 ***
## purposevacation                        0.12626
## purposewedding                         0.51371
## inq_last_6mths            < 0.0000000000000002 ***
## pub_rec                      0.000000015864915 ***
## revol_bal                              0.00315 **
## revol_util                   0.000001117244669 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22112  on 27278  degrees of freedom
## Residual deviance: 20497  on 27258  degrees of freedom
## AIC: 20539
##
## Number of Fisher Scoring iterations: 5
```

```
Anova(fit.logit.4)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: default
##                LR Chisq Df           Pr(>Chisq)
## term              172.6  1 < 0.0000000000000002 ***
## int_rate          257.6  1 < 0.0000000000000002 ***
## annual_inc        172.9  1 < 0.0000000000000002 ***
## purpose           158.3 13 < 0.0000000000000002 ***
## inq_last_6mths     66.9  1  0.00000000000000028 ***
## pub_rec            29.9  1  0.00000004491645114 ***
## revol_bal           8.4  1               0.0037 **
## revol_util         23.8  1  0.00000105530820137 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# rmse
pred4 = predict(fit.logit.4, x_test_default, type='response')
error4 = rmse(unlist(y_test_default), pred4)
error4
```

```
## [1] 0.3372
```

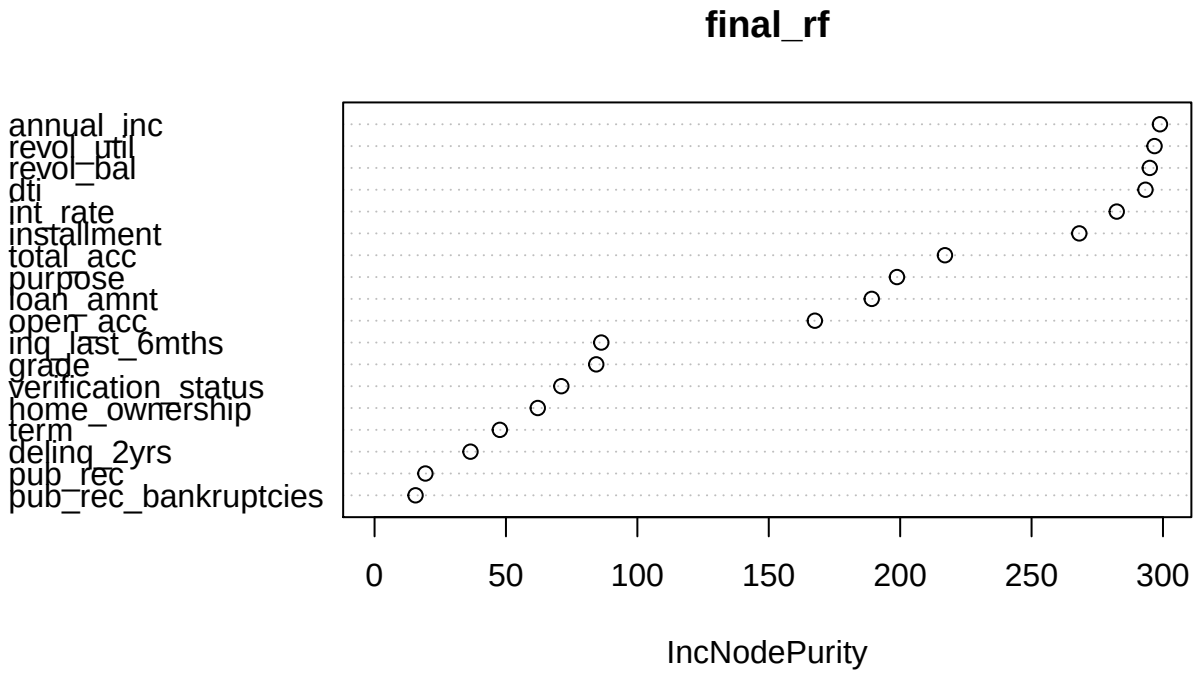
Model 3: Random Forest Regressor

```
# commenting out to Knit because it takes too long to run
#ntree = list(300, 400, 500, 600, 700)
#errors_p3 = list()


# testing errors for different values of ntrees
#length = length(ntree)
#for (i in 1:length) {
#  rf = randomForest(default ~ ., data=loan_default_train, n_tree = ntree[[i]], type="regression")

#  fit.pred = predict(rf, loan_default_test)
#  error = rmse(loan_default_test$default, fit.pred)

#  addition = cbind(ntree[[i]], error)
#  errors_p3 = rbind(errors_p3, addition)
#}
```

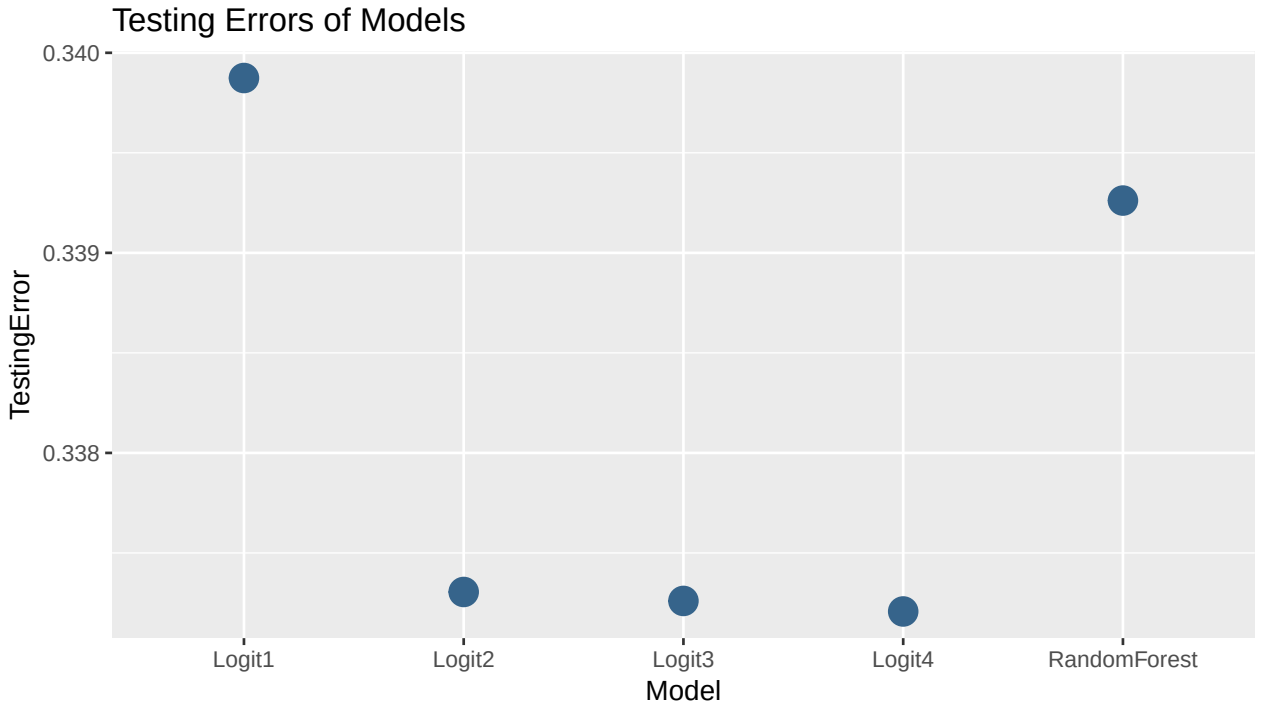
```
# commented out when Knitting because it takes too long to run
# plot errors for different values of ntrees
#df = data.frame('Trees' = unlist(errors_p3[,1]), 'TestingError' = unlist(errors_p3[,2]))
#ggplot(df, aes(x=Trees, y=TestingError)) + geom_point(col='#36648B', size=5) + ggtitle('Testing Errors
#ggsave('trees.png')
```

Based on this plot, building a random forest with 500 trees yields the lowest testing error.

```
# best rf based on testing error
final_rf = randomForest(default ~ ., data=loan_default_train, n_tree = 300, type='regression')

#fit.pred4 = ifelse(predict(final_rf, x_test_default) >= 0.5, "1", "0")
#cm4 = table(fit.pred4, unlist(y_test_default)) # confusion matrix:
#error4 = (cm4[1,2]+cm4[2,1])/length(fit.pred4)
#error4

# rmse
pred5 = predict(final_rf, x_test_default)
error5 = rmse(unlist(y_test_default), pred5)

# most important variables
varImpPlot(final_rf)
```




```
# plot testing errors of all modelsr
columns = c('Logit1', 'Logit2', 'Logit3', 'Logit4', 'RandomForest')
error_values = c(error1, error2, error3, error4, error5)

df = data.frame('Model' = columns, 'TestingError' = unlist(error_values))
ggplot(df, aes(x=Model, y=TestingError)) + geom_point(col='#36648B', size=5) + ggtitle('Testing Errors
```




```
#ggsave('models_default.png')
```

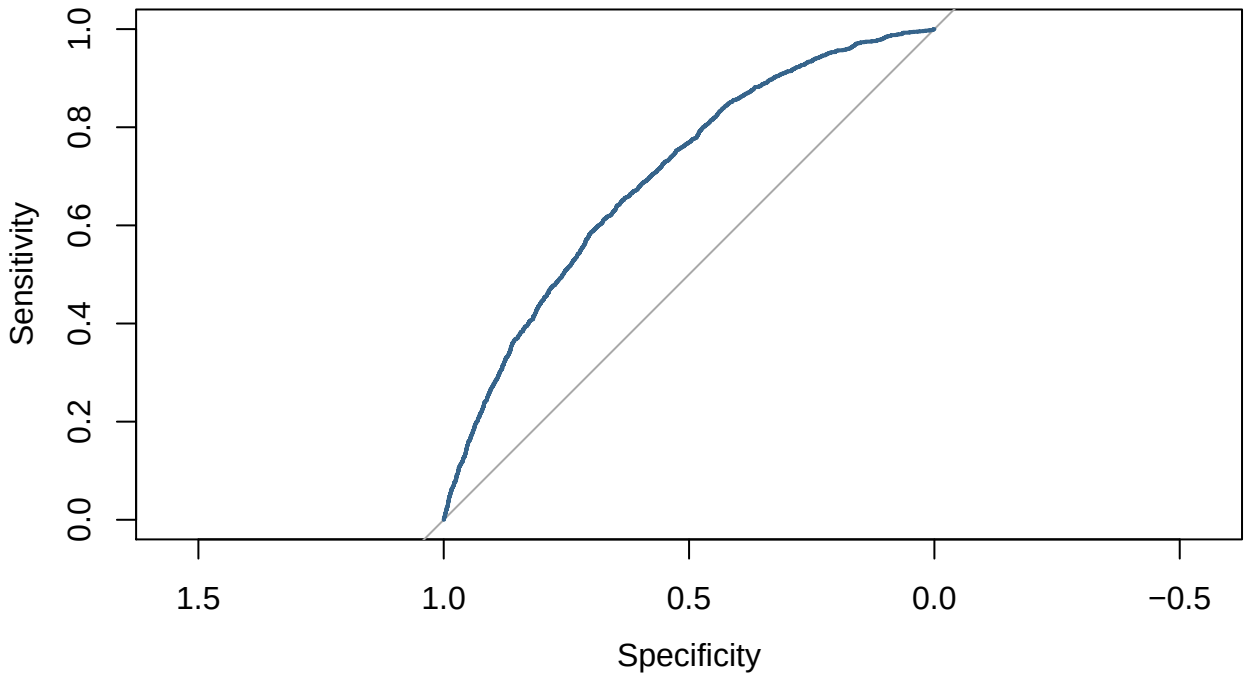
Logit4 appears to have performed the best out of all.

Goal 2: Predict which loans will default

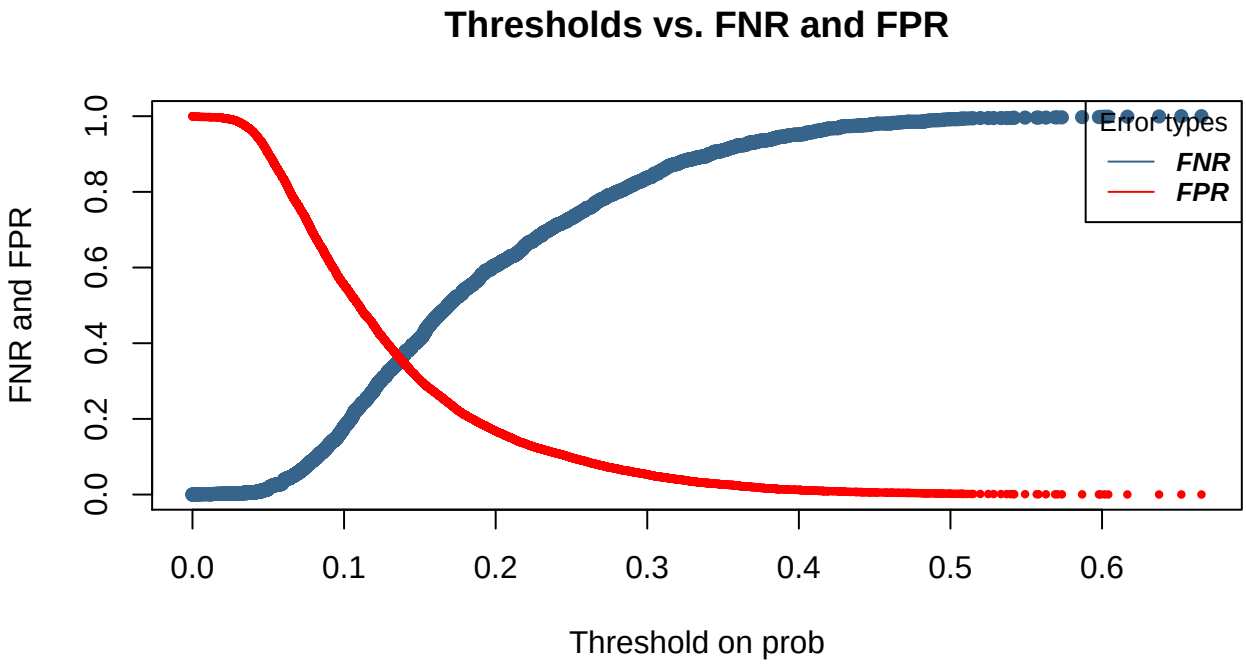
I will use fit.logit.4 from the previous part of the analysis, but will decide on which probability threshold to choose to minimize false

As an investor, I would want to minimize false negatives because I'd prefer not investing in loans that ended up not defaulting than investing in loans thatI thought would not default but ended up defaulting. However, at the same time, I don't want to give up on investments that I think will default but actually do not.

```
# roc of test date
fit.logit.roc = roc(loan_default_test$default, pred4, plot=T, col="#36648B")
```




```
# FNR and FPR
plot(fit.logit.roc$thresholds, 1-fit.logit.roc$sensitivities,  col="#36648B", pch=16,
     xlab="Threshold on prob",
     ylab="FNR and FPR",
     main = "Thresholds vs. FNR and FPR")
legend('topright', legend=c("FNR", "FPR"),
       col=c("#36648B", "red"), lty=1, cex=0.8,
       title="Error types", text.font=4)
points(fit.logit.roc$thresholds, 1-fit.logit.roc$specificities, col="red", pch=16, cex=.6)
locator()
```




```
# misclassification error with this threshold
fit.pred = ifelse(predict(fit.logit.4, x_test_default) >= 0.14, "1", "0")
cm = table(fit.pred, unlist(y_test_default)) # confusion matrix:
mis_rate = (cm[1,2]+cm[2,1])/length(fit.pred)
mis_rate
```

```
## [1] 0.141
```

Goal 3: Predict how early/late loans will be paid off

```
# create dataframe for this analysis
loan_time = loan %>% filter(default == 0) %>% select(-emp_title, -issue_d, -loan_status, -funded_amnt,

# convert difference in time to numeric
loan_time$difference_d = as.numeric(loan_time$difference_d)

# train test split
set.seed(471)
data_length = dim(loan_time)[1]
propor = sample(1:data_length, 0.7*data_length)
loan_time_train = loan_time[propor,]
loan_time_test = loan_time[-propor,]

x_time_train = loan_time_train %>% select(-difference_d)
x_time_test = loan_time_test %>% select(-difference_d)

y_time_train = loan_time_train$difference_d
y_time_test = loan_time_test$difference_d

# check the number of people who paid early/late
sum(loan_time$difference_d > 0) # paid early
```

```
## [1] 19831
```

```
sum(loan_time$difference_d < 0) # paid late
```

```
## [1] 6632
```

```
sum(loan_time$difference_d == 0) # paid on time
```

```
## [1] 7040
```

```
str(loan_time)
```

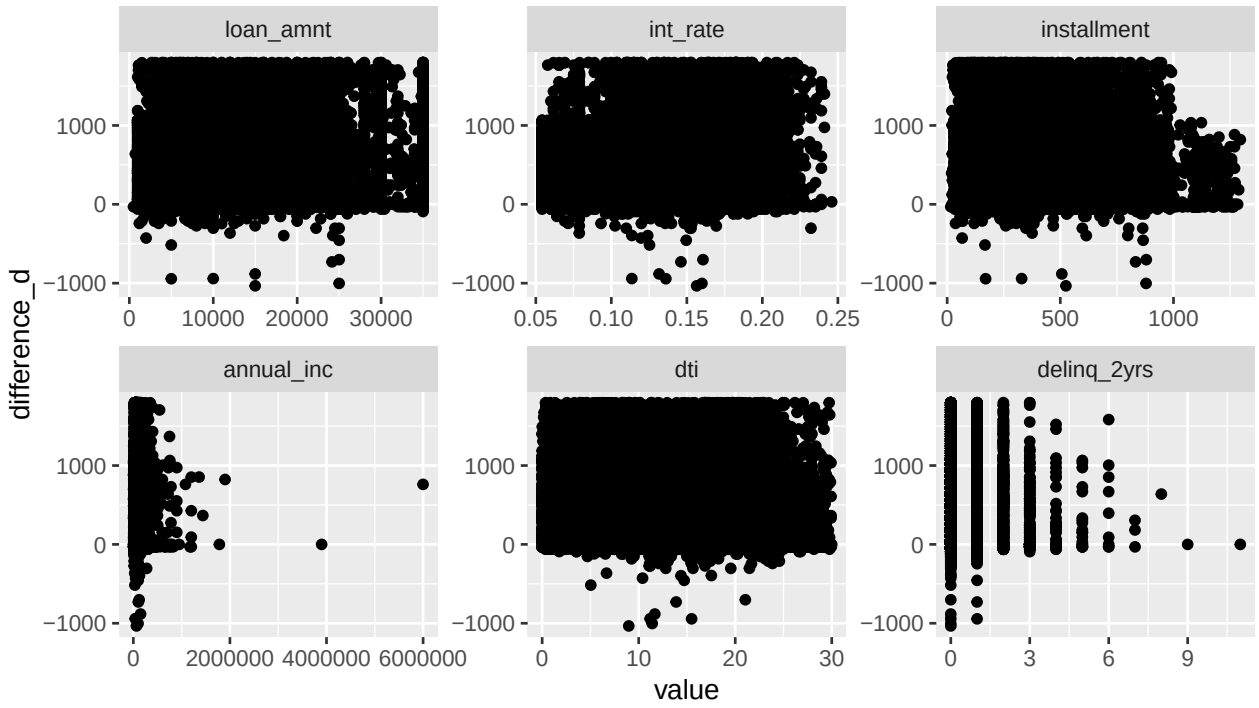
```
## 'data.frame':    33503 obs. of  19 variables:
##  $ loan_amnt           : int  5000 2400 10000 3000 5000 7000 3000 6500 12000 3000 ...
##  $ term                : num  36 36 36 60 36 60 36 60 36 36 ...
##  $ int_rate            : num  0.106 0.16 0.135 0.127 0.079 ...
##  $ installment         : num  162.9 84.3 339.3 67.8 156.5 ...
##  $ grade               : Factor w/ 7 levels "A","B","C","D",..: 2 3 3 2 1 3 5 3 2 2 ...
##  $ home_ownership      : Factor w/ 4 levels "MORTGAGE","OTHER",..: 4 4 4 4 4 4 4 4 3 3 4 ...
##  $ annual_inc          : num  24000 12252 49200 80000 36000 ...
##  $ verification_status : Factor w/ 3 levels "Not Verified",..: 3 1 2 2 2 1 2 1 2 2 ...
##  $ purpose             : Factor w/ 14 levels "car","credit_card",..: 2 12 10 10 14 3 1 3 3 2 ...
##  $ dti                 : num  27.65 8.72 20 17.94 11.2 ...
##  $ delinq_2yrs         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ inq_last_6mths      : int  1 2 1 0 3 1 2 2 0 2 ...
##  $ open_acc            : int  3 2 10 15 9 7 4 14 12 11 ...
##  $ pub_rec             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ revol_bal           : int  13648 2956 5598 27783 7963 17726 8221 4032 23336 7323 ...
##  $ revol_util          : num  0.837 0.985 0.21 0.539 0.283 0.856 0.875 0.206 0.671 0.431 ...
##  $ total_acc           : int  9 10 37 38 12 11 4 23 34 11 ...
##  $ pub_rec_bankruptcies: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ difference_d        : num  -31 183 -31 -31 -31 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
library(reshape2)
```

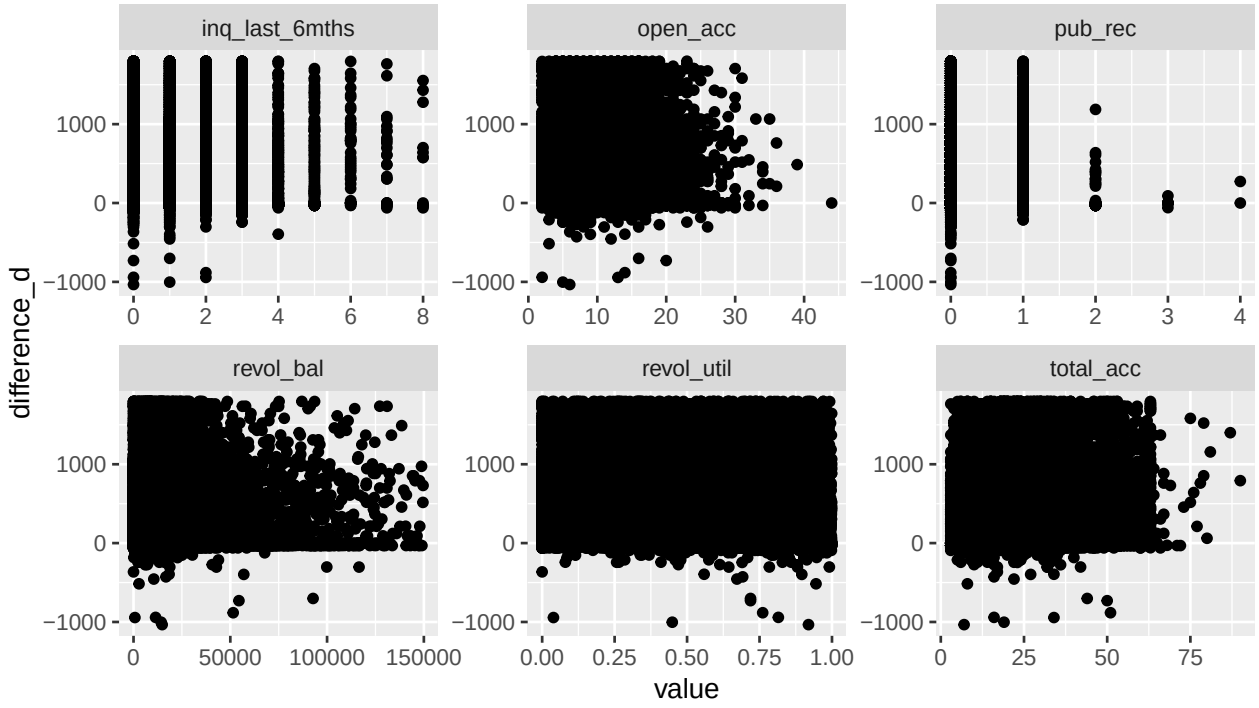
```
##
## Attaching package: 'reshape2'
```

```
## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

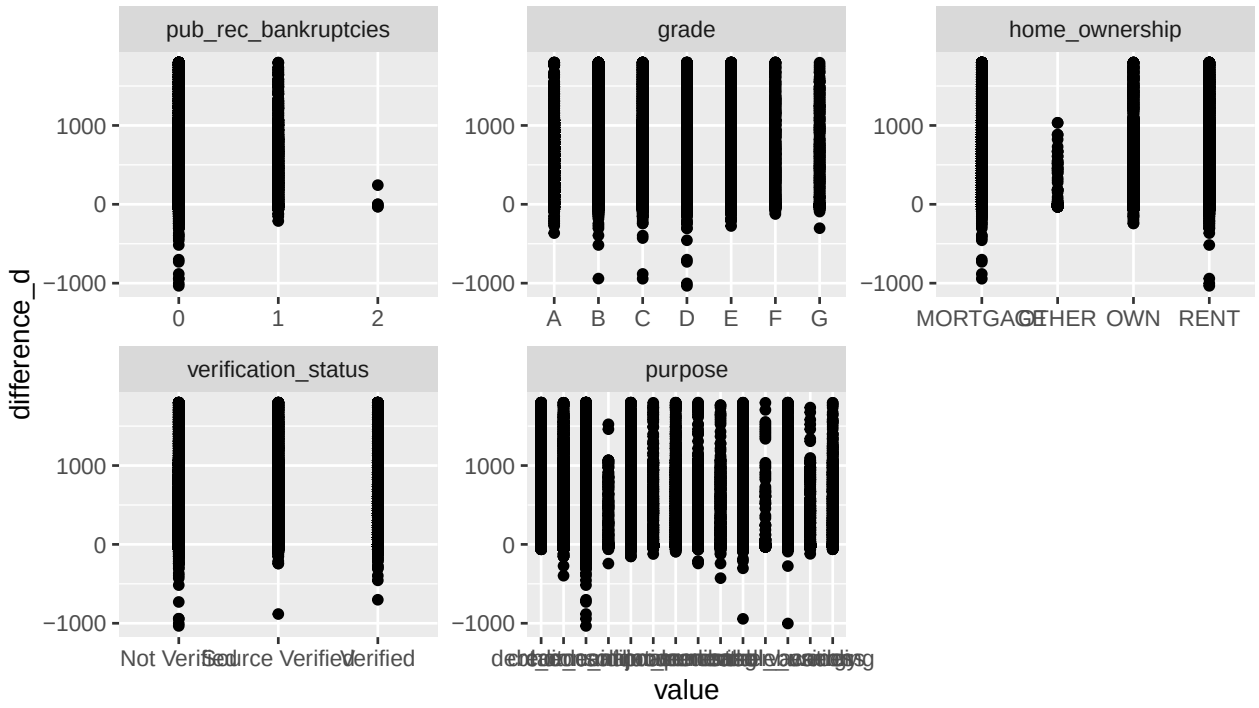
```
# plots for EDA
first = loan_time %>% select(difference_d, loan_amnt, int_rate, installment, annual_inc, dti, delinq_2y
second = loan_time %>% select(difference_d, inq_last_6mths, open_acc, pub_rec, revol_bal, revol_util, t
third = loan_time %>% select(difference_d, pub_rec_bankruptcies, grade, home_ownership, verification_st

df1 = melt(first, 'difference_d')
df2 = melt(second, 'difference_d')
df3 = melt(third, 'difference_d')

ggplot(df1, aes(value, difference_d)) +
  geom_point() +
  facet_wrap(~variable, scales = "free")
```



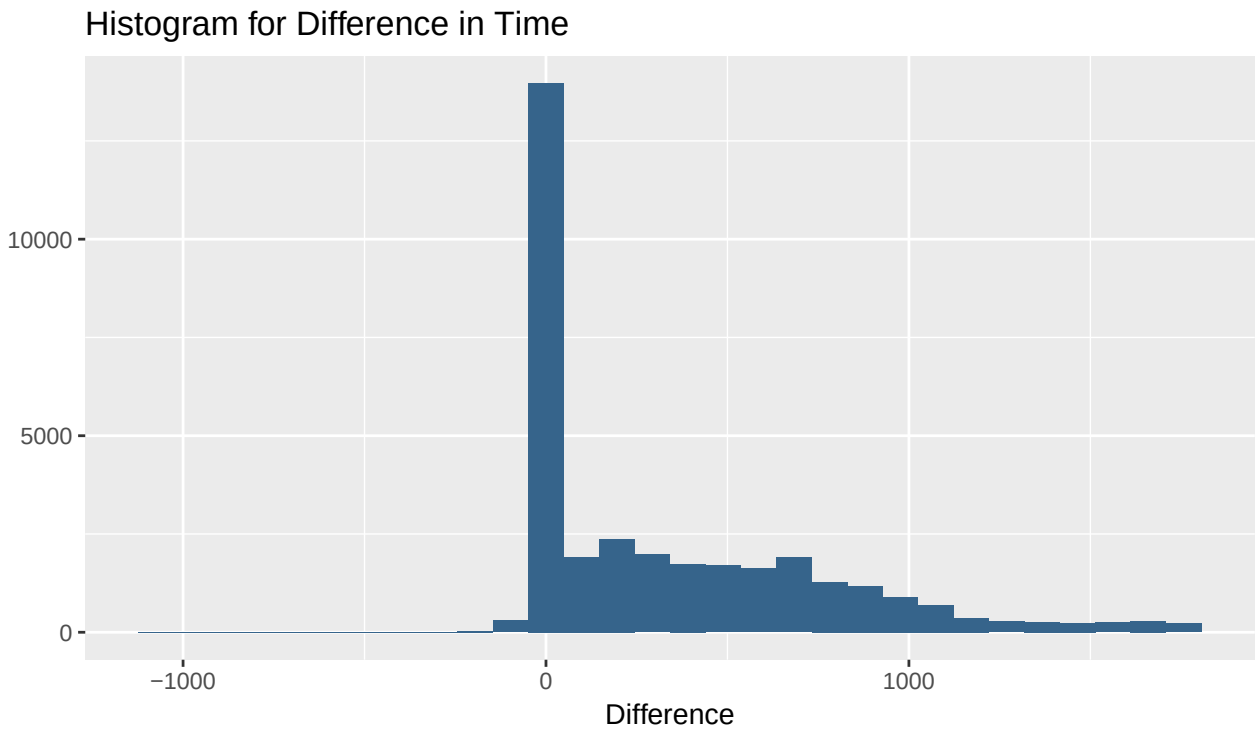

```
ggplot(df2, aes(value, difference_d)) +
  geom_point() +
  facet_wrap(~variable, scales = "free")
```




```
ggplot(df3, aes(value, difference_d)) +
  geom_point() +
  facet_wrap(~variable, scales = "free")
```




```
# histogram of difference in dates
qplot(loan_time$difference_d,
      geom="histogram",
      bins=30,
      main="Histogram for Difference in Time",
      xlab="Difference",
      fill=I("#36648B"))
```

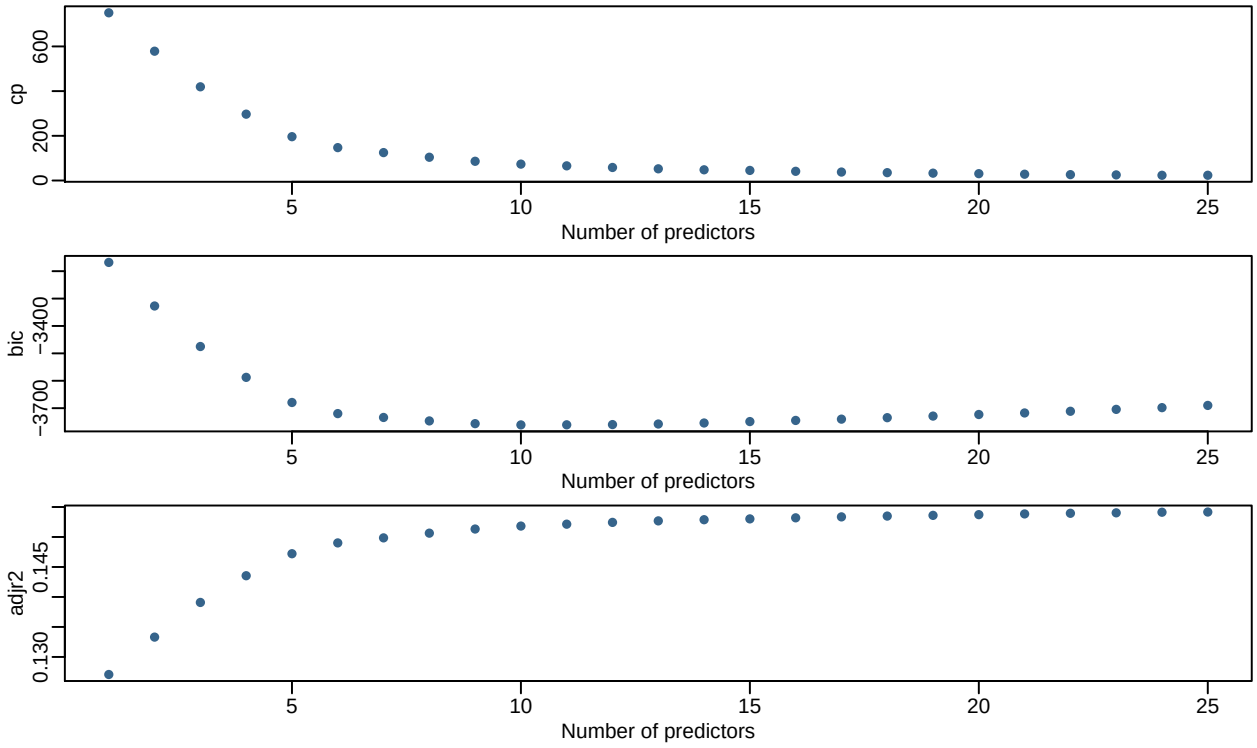


```
#ggsave('Histogram.png')
```

Model 1: Exhaustive

```
# fit exhaustive search model and choose best number of p
fit.exh = regsubsets(difference_d ~., loan_time_train, nvmax=25, method="exhaustive", really.big = T)
f.e = summary(fit.exh)
```

```
# plot cp,bic and adj R^2
par(mfrow=c(3,1), mar=c(2.5,4,0.5,1), mgp=c(1.5,0.5,0))
plot(f.e$cp, xlab="Number of predictors",
     ylab="cp", col="#36648B", type="p", pch=16)
plot(f.e$bic, xlab="Number of predictors",
     ylab="bic", col="#36648B", type="p", pch=16)
plot(f.e$adjr2, xlab="Number of predictors",
     ylab="adjr2", col="#36648B", type="p", pch=16)
```



The optimal cp model has all the variables, so I'll limit the model to include 10

```
# variables in best exhaustive search with 10 variables
fit.exh.var = f.e$which
colnames(fit.exh.var)[fit.exh.var[10,]]
```

```
##  [1] "(Intercept)"              "term"
##  [3] "int_rate"                 "purposedebt_consolidation"
##  [5] "purposesmall_business"    "dti"
##  [7] "delinq_2yrs"              "inq_last_6mths"
##  [9] "open_acc"                 "revol_util"
## [11] "total_acc"
```

```r
# fit model with those variables
fit.linear.1 = lm(difference_d ~ term + int_rate + purpose + dti + delinq_2yrs + inq_last_6mths + open_a
summary(fit.linear.1)
```

```
##
## Call:
## lm(formula = difference_d ~ term + int_rate + purpose + dti +
##     delinq_2yrs + inq_last_6mths + open_acc + revol_util + total_acc,
##     data = loan_time_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -1261   -264   -133    265   1315
##
## Coefficients:
##                            Estimate Std. Error t value             Pr(>|t|)
## (Intercept)                -229.368     17.792  -12.89 < 0.0000000000000002
## term                         13.216      0.286   46.15 < 0.0000000000000002
## int_rate                    931.220     95.679    9.73 < 0.0000000000000002
## purposecredit_card           -8.320     14.490   -0.57               0.5659
## purposedebt_consolidation    -2.185     13.204   -0.17               0.8686
## purposeeducational          -36.343     32.666   -1.11               0.2659
## purposehome_improvement     -43.341     15.545   -2.79               0.0053
## purposehouse                 36.119     29.792    1.21               0.2254
## purposemajor_purchase       -18.936     16.355   -1.16               0.2470
## purposemedical              -47.809     23.020   -2.08               0.0378
## purposemoving               -50.859     24.986   -2.04               0.0418
## purposeother                -39.463     15.017   -2.63               0.0086
## purposerenewable_energy     -23.308     52.148   -0.45               0.6549
## purposesmall_business       -77.675     17.907   -4.34   0.0000144535819756
## purposevacation              -2.742     28.651   -0.10               0.9238
## purposewedding              -19.485     20.617   -0.95               0.3446
## dti                          -3.328      0.428   -7.78   0.0000000000000073
## delinq_2yrs                 -25.382      5.435   -4.67   0.0000030292633274
## inq_last_6mths               11.841      2.524    4.69   0.0000027294679372
## open_acc                    -10.562      0.844  -12.52 < 0.0000000000000002
## revol_util                  -158.729     11.564  -13.73 < 0.0000000000000002
## total_acc                      5.788      0.318   18.21 < 0.0000000000000002
##
## (Intercept)                ***
## term                       ***
## int_rate                   ***
## purposecredit_card
## purposedebt_consolidation
## purposeeducational
## purposehome_improvement    **
## purposehouse
## purposemajor_purchase
## purposemedical             *
## purposemoving              *
## purposeother               **
## purposerenewable_energy
## purposesmall_business      ***
## purposevacation
```

```
## purposewedding
## dti                         ***
## delinq_2yrs                 ***
## inq_last_6mths              ***
## open_acc                    ***
## revol_util                  ***
## total_acc                   ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 392 on 23430 degrees of freedom
## Multiple R-squared:  0.153,  Adjusted R-squared:  0.152
## F-statistic:  202 on 21 and 23430 DF,  p-value: <0.0000000000000002
```

```
# check significance
Anova(fit.linear.1)
```

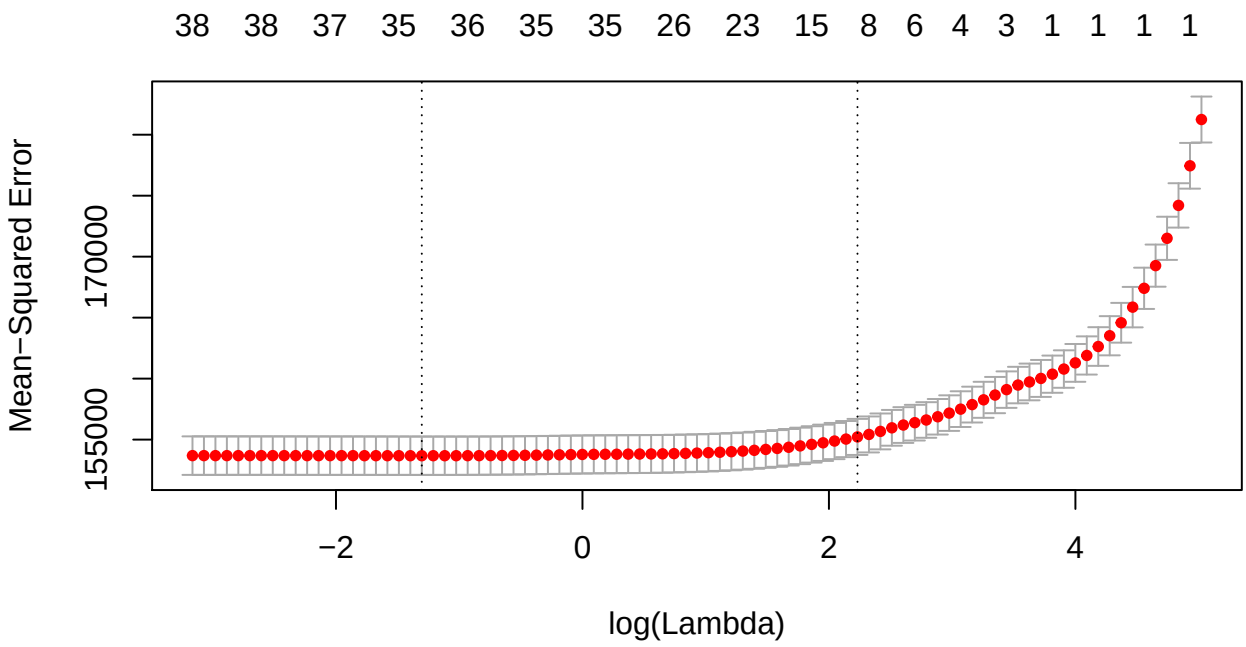
```
## Anova Table (Type II tests)
##
## Response: difference_d
##                     Sum Sq    Df F value                Pr(>F)
## term            327396510     1 2129.88 < 0.0000000000000002 ***
## int_rate         14561089     1   94.73 < 0.0000000000000002 ***
## purpose           9768941    13    4.89   0.0000000124291334 ***
## dti               9313815     1   60.59   0.0000000000000073 ***
## delinq_2yrs       3352321     1   21.81   0.0000030292633274 ***
## inq_last_6mths    3383091     1   22.01   0.0000027294679372 ***
## open_acc         24102074     1  156.80 < 0.0000000000000002 ***
## revol_util       28959942     1  188.40 < 0.0000000000000002 ***
## total_acc        50970691     1  331.59 < 0.0000000000000002 ***
## Residuals      3601565527 23430
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# test error
pred5 = predict(fit.linear.1, x_time_test)
error5 = rmse(unlist(y_time_test), pred5)
error5
```

```
## [1] 391.5
```

Model 2: LASSO Regression

```
# data prep
x_train = model.matrix(loan_time_train$difference_d~., data=loan_time_train)[,-1]
x_test = model.matrix(loan_time_test$difference_d~., data=loan_time_test)[,-1]

# cross validation to select lambda
fit.time.cv = cv.glmnet(x_train, y_time_train, alpha=1, nfolds=10 )
plot(fit.time.cv)
```




```
# use lambda that returns min value of mse
coef.min = coef(fit.time.cv, s="lambda.1se")
coef.min = coef.min[which(coef.min!=0),]
var.min = rownames(as.matrix(coef.min))
var.min
```

```
##  [1] "(Intercept)"                "loan_amnt"
##  [3] "term"                       "int_rate"
##  [5] "verification_statusVerified" "dti"
##  [7] "inq_last_6mths"             "open_acc"
##  [9] "revol_util"                 "total_acc"
```

```
# fit lm model
fit.linear.2 = lm(difference_d ~ loan_amnt + term + int_rate + verification_status + dti + inq_last_6mt
summary(fit.linear.2)
```

```
##
## Call:
## lm(formula = difference_d ~ loan_amnt + term + int_rate + verification_status +
##     dti + inq_last_6mths + open_acc + revol_util + total_acc,
##     data = loan_time_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -1242   -264   -137    268   1304
##
## Coefficients:
##                                    Estimate  Std. Error t value
## (Intercept)                     -249.499352   13.158728  -18.96
## loan_amnt                          0.000796    0.000421    1.89
## term                              13.183195    0.291691   45.20
## int_rate                         712.391489   94.069681    7.57
## verification_statusSource Verified   7.225666    6.496888    1.11
## verification_statusVerified         16.392848    6.677827    2.45
## dti                                 -3.013542    0.425526   -7.08
```

```
## inq_last_6mths                        12.204683    2.521004    4.84
## open_acc                              -9.894922    0.839672  -11.78
## revol_util                          -136.651458   11.198552  -12.20
## total_acc                              5.373299    0.322092   16.68
##                                                        Pr(>|t|)
## (Intercept)                         < 0.0000000000000002 ***
## loan_amnt                                          0.059 .
## term                                < 0.0000000000000002 ***
## int_rate                               0.000000000000038 ***
## verification_statusSource Verified                 0.266
## verification_statusVerified                        0.014 *
## dti                                    0.000000000001462 ***
## inq_last_6mths                         0.000001298788732 ***
## open_acc                            < 0.0000000000000002 ***
## revol_util                          < 0.0000000000000002 ***
## total_acc                           < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 393 on 23441 degrees of freedom
## Multiple R-squared:  0.151,  Adjusted R-squared:  0.15
## F-statistic:  415 on 10 and 23441 DF,  p-value: <0.0000000000000002
```

```
# test error
pred6 = predict(fit.linear.2, x_time_test)
error6 = rmse(unlist(y_time_test), pred6)
error6
```

```
## [1] 391.7
```

Remove insignificant variables

```
Anova(fit.linear.2)
```

```
## Anova Table (Type II tests)
##
## Response: difference_d
##                          Sum Sq    Df F value                Pr(>F)
## loan_amnt                 550211     1    3.57                 0.059 .
## term                   314791573     1 2042.66 < 0.0000000000000002 ***
## int_rate                 8838238     1   57.35     0.000000000000038 ***
## verification_status       930584     2    3.02                 0.049 *
## dti                      7729104     1   50.15     0.000000000001462 ***
## inq_last_6mths           3611883     1   23.44     0.000001298788732 ***
## open_acc                21400966     1  138.87 < 0.0000000000000002 ***
## revol_util              22947340     1  148.90 < 0.0000000000000002 ***
## total_acc               42889296     1  278.31 < 0.0000000000000002 ***
## Residuals             3612465490 23441
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# fit lm model
fit.linear.3 = lm(difference_d ~ term + int_rate + dti + inq_last_6mths + open_acc + revol_util + total_
summary(fit.linear.3)
```

```
##
## Call:
```

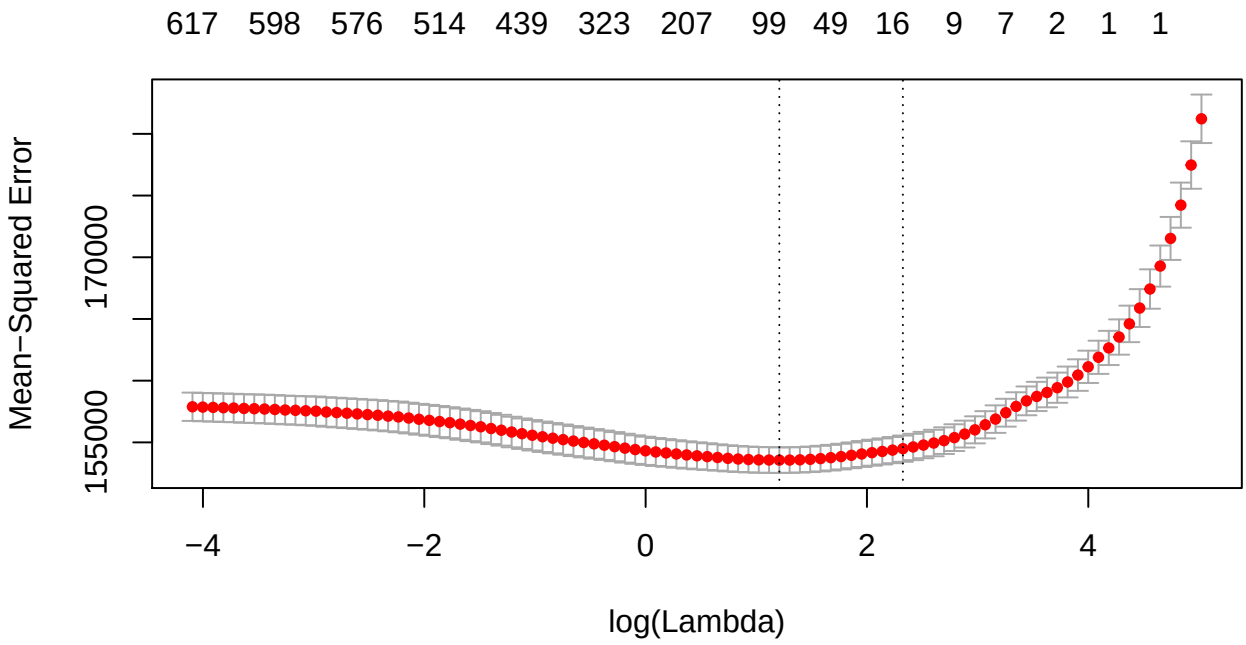
```
## lm(formula = difference_d ~ term + int_rate + dti + inq_last_6mths +
##     open_acc + revol_util + total_acc, data = loan_time_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -1243   -264   -136    268   1316
##
## Coefficients:
##                Estimate Std. Error t value             Pr(>|t|)
## (Intercept)    -254.603     12.993  -19.60 < 0.0000000000000002 ***
## term             13.427      0.283   47.45 < 0.0000000000000002 ***
## int_rate        775.538     92.172    8.41 < 0.0000000000000002 ***
## dti              -3.025      0.424   -7.13        0.000000000001 ***
## inq_last_6mths   11.680      2.514    4.65        0.000003407968 ***
## open_acc         -9.884      0.840  -11.77 < 0.0000000000000002 ***
## revol_util     -137.587     11.191  -12.29 < 0.0000000000000002 ***
## total_acc         5.569      0.315   17.67 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 393 on 23444 degrees of freedom
## Multiple R-squared:  0.15,   Adjusted R-squared:  0.15
## F-statistic:  591 on 7 and 23444 DF,  p-value: <0.0000000000000002
```

```
# test error
pred7 = predict(fit.linear.3, x_time_test)
error7 = rmse(unlist(y_time_test), pred7)
error7
```

```
## [1] 391.7
```

Model 3: Interaction terms

```
# generate data
interaction_data = model.matrix(difference_d ~.^2, data=loan_time)
x_inter_train = interaction_data[propor,]
x_inter_test = interaction_data[-propor,]

# cross validation to select lambda
fit.inter.cv = cv.glmnet(x_inter_train, y_time_train, alpha=1, nfolds=10 )
plot(fit.inter.cv)
```




```
# variables for lambda 1se
coef.1se = coef(fit.inter.cv, s="lambda.1se")
coef.1se = coef.1se[which(coef.1se!=0),]
var.1se = rownames(as.matrix(coef.1se))
var.1se
```

```
##  [1] "(Intercept)"
##  [2] "term"
##  [3] "dti"
##  [4] "revol_util"
##  [5] "loan_amnt:term"
##  [6] "term:int_rate"
##  [7] "term:verification_statusVerified"
##  [8] "term:inq_last_6mths"
##  [9] "term:total_acc"
## [10] "purposedebt_consolidation:inq_last_6mths"
## [11] "dti:open_acc"
## [12] "open_acc:revol_util"
```

```
#data = data.frame(cbind(y_time_train, x_inter_train))

fit.linear.4 = lm(difference_d ~ term + dti + revol_util + loan_amnt*term + term*int_rate + term*verifi
summary(fit.linear.4)
```

```
##
## Call:
## lm(formula = difference_d ~ term + dti + revol_util + loan_amnt *
##     term + term * int_rate + term * verification_status + term *
##     inq_last_6mths + term * total_acc + dti * open_acc + open_acc *
##     revol_util, data = loan_time_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -1209   -266   -138    266   1355
##
```

```
## Coefficients:
##                                          Estimate    Std. Error
## (Intercept)                            213.4060786    50.8611969
## term                                     1.8547218     1.1783631
## dti                                     -2.8717617     0.9208628
## revol_util                            -108.6574867    21.8150335
## loan_amnt                               -0.0026584     0.0016972
## int_rate                              -1318.7675260   336.2688524
## verification_statusSource Verified       -6.1241221    28.6323359
## verification_statusVerified             -54.1461055    29.1400452
## inq_last_6mths                          -15.5928575    10.1974384
## total_acc                                0.1870164     1.0105055
## open_acc                                -8.2843245     1.6652774
## term:loan_amnt                           0.0000702     0.0000374
## term:int_rate                           46.6379231     7.5321116
## term:verification_statusSource Verified     0.3591920     0.6843226
## term:verification_statusVerified            1.6565613     0.6922817
## term:inq_last_6mths                         0.6510037     0.2319883
## term:total_acc                              0.1211459     0.0225852
## dti:open_acc                               -0.0125941     0.0926283
## revol_util:open_acc                        -2.7745339     2.2189895
##                                        t value     Pr(>|t|)
## (Intercept)                               4.20 0.0000272848 ***
## term                                      1.57       0.1155
## dti                                      -3.12       0.0018 **
## revol_util                               -4.98 0.0000006375 ***
## loan_amnt                                -1.57       0.1173
## int_rate                                 -3.92 0.0000881529 ***
## verification_statusSource Verified       -0.21       0.8306
## verification_statusVerified              -1.86       0.0632 .
## inq_last_6mths                           -1.53       0.1263
## total_acc                                 0.19       0.8532
## open_acc                                 -4.97 0.0000006580 ***
## term:loan_amnt                            1.88       0.0607 .
## term:int_rate                             6.19 0.0000000006 ***
## term:verification_statusSource Verified   0.52       0.5997
## term:verification_statusVerified          2.39       0.0167 *
## term:inq_last_6mths                       2.81       0.0050 **
## term:total_acc                            5.36 0.0000000822 ***
## dti:open_acc                             -0.14       0.8919
## revol_util:open_acc                      -1.25       0.2112
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 392 on 23433 degrees of freedom
## Multiple R-squared:  0.155,  Adjusted R-squared:  0.155
## F-statistic:  239 on 18 and 23433 DF,  p-value: <0.0000000000000002
```

```
# test error
pred8 = predict(fit.linear.4, x_time_test)
error8 = rmse(unlist(y_time_test), pred8)
error8
```

```
## [1] 391.5
```

```r
Anova(fit.linear.4)
```

```
## Anova Table (Type II tests)
##
## Response: difference_d
##                               Sum Sq    Df F value                Pr(>F)
## term                       314139089     1 2048.84 < 0.0000000000000002 ***
## dti                          7551623     1   49.25     0.00000000000231 ***
## revol_util                  21313912     1  139.01 < 0.0000000000000002 ***
## loan_amnt                     153717     1    1.00                0.317
## int_rate                     7901539     1   51.53     0.00000000000072 ***
## verification_status           735424     2    2.40                0.091 .
## inq_last_6mths               3566917     1   23.26     0.00000142125923 ***
## total_acc                   42097696     1  274.56 < 0.0000000000000002 ***
## open_acc                    20409133     1  133.11 < 0.0000000000000002 ***
## term:loan_amnt                539551     1    3.52                0.061 .
## term:int_rate                5878396     1   38.34     0.00000000060439 ***
## term:verification_status     1024113     2    3.34                0.035 *
## term:inq_last_6mths          1207393     1    7.87                0.005 **
## term:total_acc               4411481     1   28.77     0.00000008218991 ***
## dti:open_acc                    2834     1    0.02                0.892
## revol_util:open_acc           239709     1    1.56                0.211
## Residuals                 3592872294 23433
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#remove insignificant terms

fit.linear.5 = lm(y_time_train ~ term + dti + revol_util + term*int_rate + term*verification_status + te
summary(fit.linear.5)
```

```
##
## Call:
## lm(formula = y_time_train ~ term + dti + revol_util + term *
##     int_rate + term * verification_status + term * inq_last_6mths +
##     term * total_acc, data = loan_time_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -1220   -267   -141    269   1375
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                              200.1531    49.5374    4.04
## term                                       2.1144     1.1784    1.79
## dti                                       -4.1187     0.4145   -9.94
## revol_util                              -111.1261    11.0608  -10.05
## int_rate                               -1554.7135   331.9109   -4.68
## verification_statusSource Verified        -14.0470    28.5451   -0.49
## verification_statusVerified               -79.4943    27.0230   -2.94
## inq_last_6mths                            -15.3200    10.2001   -1.50
## total_acc                                  -2.7784     0.9583   -2.90
## term:int_rate                              49.1925     7.4296    6.62
## term:verification_statusSource Verified     0.5364     0.6812    0.79
## term:verification_statusVerified            2.3077     0.6356    3.63
```

```
## term:inq_last_6mths                          0.6455     0.2320    2.78
## term:total_acc                               0.1359     0.0219    6.19
##                                                     Pr(>|t|)
## (Intercept)                                   0.000053519779 ***
## term                                                0.07278 .
## dti                                        < 0.0000000000000002 ***
## revol_util                                 < 0.0000000000000002 ***
## int_rate                                      0.000002827283 ***
## verification_statusSource Verified                  0.62266
## verification_statusVerified                         0.00327 **
## inq_last_6mths                                      0.13312
## total_acc                                           0.00374 **
## term:int_rate                                 0.000000000036 ***
## term:verification_statusSource Verified             0.43106
## term:verification_statusVerified                    0.00028 ***
## term:inq_last_6mths                                 0.00540 **
## term:total_acc                                0.000000000610 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 393 on 23438 degrees of freedom
## Multiple R-squared:  0.15,   Adjusted R-squared:  0.15
## F-statistic:  318 on 13 and 23438 DF,  p-value: <0.0000000000000002
```

```
Anova(fit.linear.5)
```

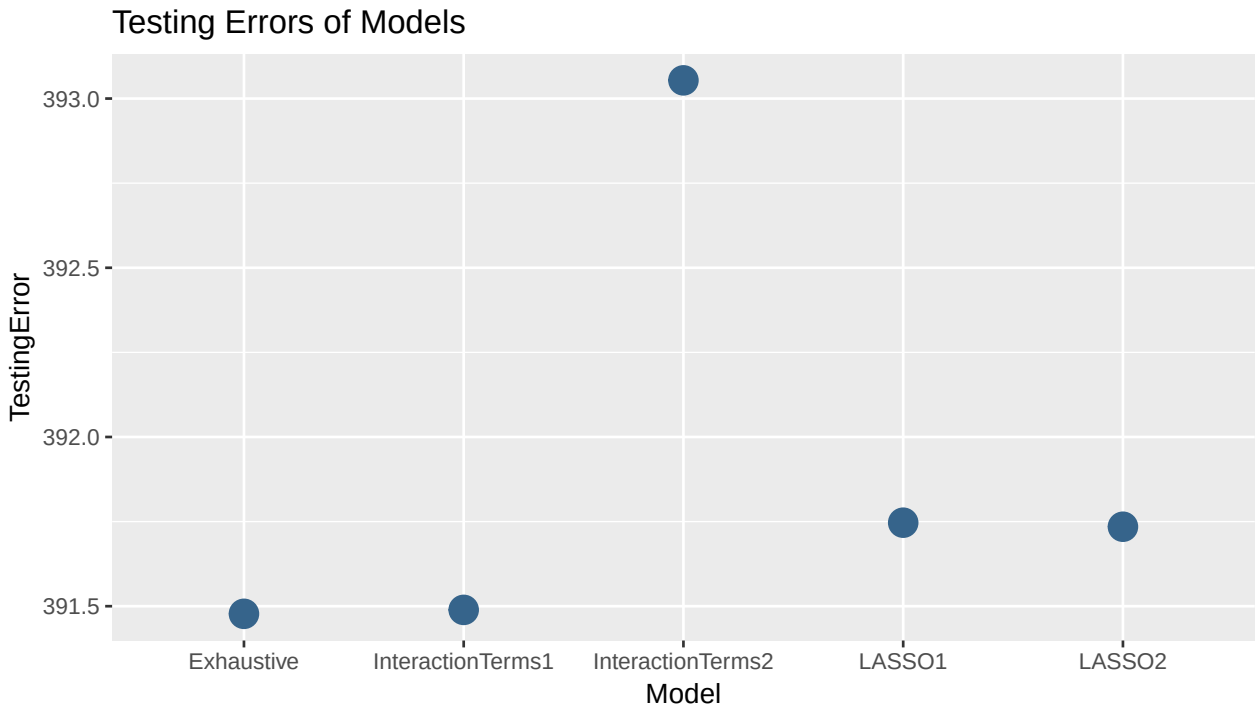
```
## Anova Table (Type II tests)
##
## Response: y_time_train
##                              Sum Sq    Df F value                Pr(>F)
## term                       346597138     1 2247.58 < 0.0000000000000002 ***
## dti                         15223878     1   98.72 < 0.0000000000000002 ***
## revol_util                  15565764     1  100.94 < 0.0000000000000002 ***
## int_rate                     5580962     1   36.19       0.000000001816 ***
## verification_status          1210727     2    3.93              0.01974 *
## inq_last_6mths               3618295     1   23.46       0.000001281111 ***
## total_acc                   23867386     1  154.77 < 0.0000000000000002 ***
## term:int_rate                6760379     1   43.84       0.000000000036 ***
## term:verification_status     2383802     2    7.73              0.00044 ***
## term:inq_last_6mths          1193719     1    7.74              0.00540 **
## term:total_acc               5909250     1   38.32       0.000000000610 ***
## Residuals                 3614345195 23438
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# test error
pred9 = predict(fit.linear.5, x_time_test)
error9 = rmse(unlist(y_time_test), pred9)
error9
```

```
## [1] 393.1
```

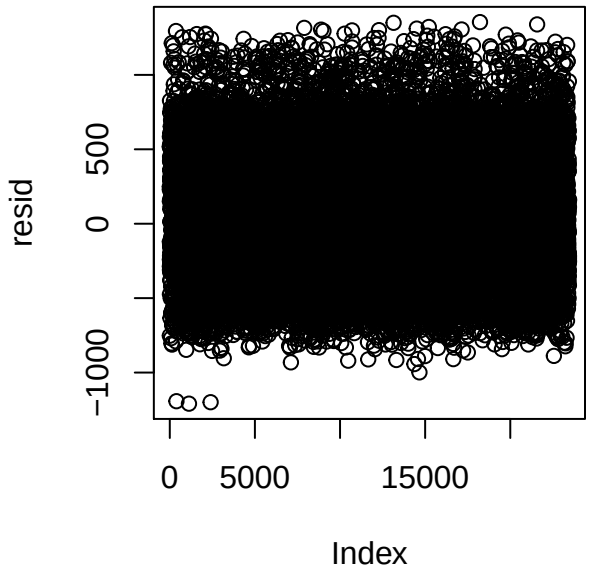
```
# plot errors of each model
columns = c('Exhaustive', 'LASSO1', 'LASSO2', 'InteractionTerms1', 'InteractionTerms2')
error_values = c(error5, error6, error7, error8, error9)
```

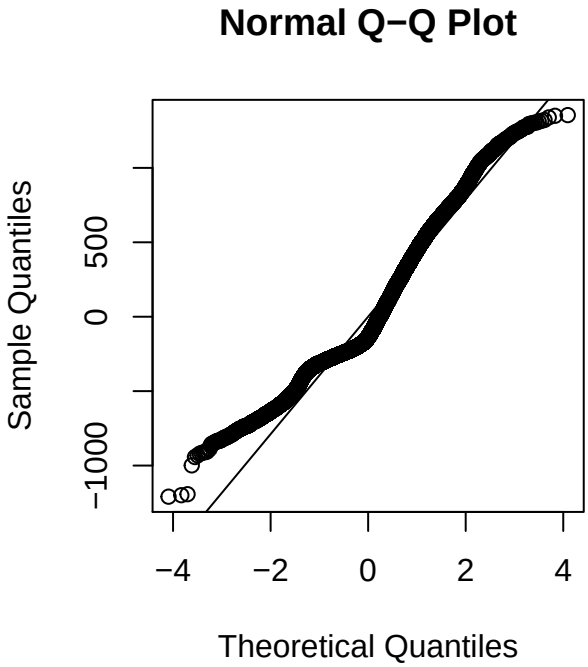
```
df = data.frame('Model' = columns, 'TestingError' = unlist(error_values))
ggplot(df, aes(x=Model, y=TestingError)) + geom_point(col='#36648B', size=5) + ggtitle('Testing Errors
```




```
#ggsave('errors_linear.png')
```

```
# check residuals
resid = resid(fit.linear.4)
par(mfrow=c(1,2))
plot(resid)
qqnorm(resid)
qqline(resid)
```


resid
−1000
0
500
0
5000
15000
Index


Normal Q−Q Plot
Sample Quantiles
−1000
0
500
−4
−2
0
2
4
Theoretical Quantiles