# Project 4: Prosper Loan Data

*Joby John*

*November 10, 2015*

## Contents



## Introduction

This dataset involves the loan data from Prosper Loans. The aim is conduct an Exploratory Data Analysis (EDA) of this dataset and find hidden relationships and important features that affect the lender and the borrower. Going through the description of the features one can get a good idea of the lender-borrower relationship. The lender wants to maximize his returns or yield from the Borrower and the latter wants to ideally pay off the debt without accruing too much interest and service fees. However due to various constraints (like limited income and prior debt) the borrower has to agree to terms which are not in his best interest or in the worst case even default on the payments; the lender on the other hand lends to borrowers whose creditworthiness is not pristine because he hopes to make a profit by levying interest and late fee on such borrower.

From a lender's perspective, he is looking to understand the risk vs benefit in lending to a certain borrower and thus looks at all factors that point to his credit worthiness. From a borrower's perspective one can understand factors that affect one's ability to pay back the loan in a timely manner without overpaying by analyzing the relationship between various variables. Going through the description of some of the variables we can identify some of the variables that are of interest to us.

We start off by looking at matrix plots and narrow down our analysis by looking at some trends displayed in these plots. We pursue the effect of credit rating, credit scores, which measure credit worthiness on different factors that determine the yield for the Lender and the payments for the borrower. We also identify trends in the variables and see how these trends are affected by various features of the borrower's profile.

#### Initialize Global Options

## Load the Data and Libraries

```
pd <- read.csv('prosperLoanData.csv')
library(reshape2)
library(RColorBrewer)
library(ggplot2)
library(extrafont) # For CM Roman fonts on figure labels.
library(GGally)
str(pd) # Structure of the dataframe
```

```
'data.frame':   113937 obs. of  81 variables:
 $ ListingKey                        : Factor w/ 113066 levels "0000354648209428 2EF90E5",..:
    7180 7193 6647 6669 6686 6689 6699 6706 6687 6687 ...
 $ ListingNumber                     : int  193129 1209647 81716 658116 909464 1074836 750899
    768193 1023355 1023355 ...
 $ ListingCreationDate               : Factor w/ 113064 levels "2005-11-09
    20:44:28.847000000",..: 14184 111894 6429 64760 85967 100310 72556 74019 97834 97834 ...
 $ CreditGrade                       : Factor w/ 9 levels "","A","AA","B",..: 5 1 8 1 1 1 1 1 1
    1 ...
 $ Term                              : int  36 36 36 36 36 60 36 36 36 36 ...
 $ LoanStatus                        : Factor w/ 12 levels "Cancelled","Chargedoff",..: 3 4 3 4
    4 4 4 4 4 4 ...
 $ ClosedDate                        : Factor w/ 2803 levels "","2005-11-25 00:00:00",..: 1138 1
     1263 1 1 1 1 1 1 ...
 $ BorrowerAPR                       : num  0.165 0.12 0.283 0.125 0.246 ...
 $ BorrowerRate                      : num  0.158 0.092 0.275 0.0974 0.2085 ...
 $ LenderYield                       : num  0.138 0.082 0.24 0.0874 0.1985 ...
 $ EstimatedEffectiveYield           : num  NA 0.0796 NA 0.0849 0.1832 ...
 $ EstimatedLoss                     : num  NA 0.0249 NA 0.0249 0.0925 ...
 $ EstimatedReturn                   : num  NA 0.0547 NA 0.06 0.0907 ...
 $ ProsperRating..numeric.           : int  NA 6 NA 6 3 5 2 4 7 7 ...
 $ ProsperRating..Alpha.             : Factor w/ 8 levels "","A","AA","B",..: 1 2 1 2 6 4 7 5 3
    3 ...
 $ ProsperScore                      : num  NA 7 NA 9 4 10 2 4 9 11 ...
 $ ListingCategory..numeric.         : int  0 2 0 16 2 1 1 2 7 7 ...
 $ BorrowerState                     : Factor w/ 52 levels "","AK","AL","AR",..: 7 7 12 12 25 34
     18 6 16 16 ...
 $ Occupation                        : Factor w/ 68 levels "","Accountant/CPA",..: 37 43 37 52
    21 43 50 29 24 24 ...
 $ EmploymentStatus                  : Factor w/ 9 levels "","Employed",..: 9 2 4 2 2 2 2 2 2 2
    ...
 $ EmploymentStatusDuration          : int  2 44 NA 113 44 82 172 103 269 269 ...
 $ IsBorrowerHomeowner               : Factor w/ 2 levels "False","True": 2 1 1 2 2 2 1 1 2 2
    ...
 $ CurrentlyInGroup                  : Factor w/ 2 levels "False","True": 2 1 2 1 1 1 1 1 1 1
    ...
```

```
$ GroupKey                          : Factor w/ 707 levels "","00343376901312423168731",..: 1 1
    335 1 1 1 1 1 1 ...
$ DateCreditPulled                  : Factor w/ 112992 levels "2005-11-09
   00:30:04.487000000",..: 14347 111883 6446 64724 85857 100382 72500 73937 97888 97888 ...
$ CreditScoreRangeLower             : int  640 680 480 800 680 740 680 700 820 820 ...
$ CreditScoreRangeUpper             : int  659 699 499 819 699 759 699 719 839 839 ...
$ FirstRecordedCreditLine           : Factor w/ 11586 levels "","1947-08-24 00:00:00",..: 8639
   6617 8927 2247 9498 497 8265 7685 5543 5543 ...
$ CurrentCreditLines                : int  5 14 NA 5 19 21 10 6 17 17 ...
$ OpenCreditLines                   : int  4 14 NA 5 19 17 7 6 16 16 ...
$ TotalCreditLinespast7years        : int  12 29 3 29 49 49 20 10 32 32 ...
$ OpenRevolvingAccounts             : int  1 13 0 7 6 13 6 5 12 12 ...
$ OpenRevolvingMonthlyPayment       : num  24 389 0 115 220 1410 214 101 219 219 ...
$ InquiriesLast6Months              : int  3 3 0 0 1 0 0 3 1 1 ...
$ TotalInquiries                    : num  3 5 1 1 9 2 0 16 6 6 ...
$ CurrentDelinquencies              : int  2 0 1 4 0 0 0 0 0 0 ...
$ AmountDelinquent                  : num  472 0 NA 10056 0 ...
$ DelinquenciesLast7Years           : int  4 0 0 14 0 0 0 0 0 0 ...
$ PublicRecordsLast10Years          : int  0 1 0 0 0 0 0 1 0 0 ...
$ PublicRecordsLast12Months         : int  0 0 NA 0 0 0 0 0 0 0 ...
$ RevolvingCreditBalance            : num  0 3989 NA 1444 6193 ...
$ BankcardUtilization               : num  0 0.21 NA 0.04 0.81 0.39 0.72 0.13 0.11 0.11 ...
$ AvailableBankcardCredit           : num  1500 10266 NA 30754 695 ...
$ TotalTrades                       : num  11 29 NA 26 39 47 16 10 29 29 ...
$ TradesNeverDelinquent..percentage. : num  0.81 1 NA 0.76 0.95 1 0.68 0.8 1 1 ...
$ TradesOpenedLast6Months           : num  0 2 NA 0 2 0 0 0 1 1 ...
$ DebtToIncomeRatio                 : num  0.17 0.18 0.06 0.15 0.26 0.36 0.27 0.24 0.25 0.25
   ...
$ IncomeRange                       : Factor w/ 8 levels "$0","$100,000+",..: 4 5 7 4 2 2 4 4 4
    4 ...
$ IncomeVerifiable                  : Factor w/ 2 levels "False","True": 2 2 2 2 2 2 2 2 2 2
   ...
$ StatedMonthlyIncome               : num  3083 6125 2083 2875 9583 ...
$ LoanKey                           : Factor w/ 113066 levels "00003683605746079487FF7",..:
   100337 69837 46303 70776 71387 86505 91250 5425 908 908 ...
$ TotalProsperLoans                 : int  NA NA NA NA 1 NA NA NA NA NA ...
$ TotalProsperPaymentsBilled        : int  NA NA NA NA 11 NA NA NA NA NA ...
$ OnTimeProsperPayments             : int  NA NA NA NA 11 NA NA NA NA NA ...
$ ProsperPaymentsLessThanOneMonthLate: int  NA NA NA NA 0 NA NA NA NA NA ...
$ ProsperPaymentsOneMonthPlusLate   : int  NA NA NA NA 0 NA NA NA NA NA ...
$ ProsperPrincipalBorrowed          : num  NA NA NA NA 11000 NA NA NA NA NA ...
$ ProsperPrincipalOutstanding       : num  NA NA NA NA 9948 ...
$ ScorexChangeAtTimeOfListing       : int  NA NA NA NA NA NA NA NA NA NA ...
$ LoanCurrentDaysDelinquent         : int  0 0 0 0 0 0 0 0 0 0 ...
$ LoanFirstDefaultedCycleNumber     : int  NA NA NA NA NA NA NA NA NA NA ...
$ LoanMonthsSinceOrigination        : int  78 0 86 16 6 3 11 10 3 3 ...
$ LoanNumber                        : int  19141 134815 6466 77296 102670 123257 88353 90051
   121268 121268 ...
$ LoanOriginalAmount                : int  9425 10000 3001 10000 15000 15000 3000 10000 10000
   10000 ...
$ LoanOriginationDate               : Factor w/ 1873 levels "2005-11-15 00:00:00",..: 426 1866
   260 1535 1757 1821 1649 1666 1813 1813 ...
$ LoanOriginationQuarter            : Factor w/ 33 levels "Q1 2006","Q1 2007",..: 18 8 2 32 24
   33 16 16 33 33 ...
$ MemberKey                         : Factor w/ 90831 levels "00003397697413387CAF966",..:
   11071 10302 33781 54939 19465 48037 60448 40951 26129 26129 ...
$ MonthlyLoanPayment                : num  330 319 123 321 564 ...
$ LP_CustomerPayments               : num  11396 0 4187 5143 2820 ...
$ LP_CustomerPrincipalPayments      : num  9425 0 3001 4091 1563 ...
```

```
$ LP_InterestandFees                 : num  1971 0 1186 1052 1257 ...
$ LP_ServiceFees                     : num  -133.2 0 -24.2 -108 -60.3 ...
$ LP_CollectionFees                  : num  0 0 0 0 0 0 0 0 0 0 ...
$ LP_GrossPrincipalLoss              : num  0 0 0 0 0 0 0 0 0 0 ...
$ LP_NetPrincipalLoss                : num  0 0 0 0 0 0 0 0 0 0 ...
$ LP_NonPrincipalRecoverypayments    : num  0 0 0 0 0 0 0 0 0 0 ...
$ PercentFunded                      : num  1 1 1 1 1 1 1 1 1 1 ...
$ Recommendations                    : int  0 0 0 0 0 0 0 0 0 0 ...
$ InvestmentFromFriendsCount         : int  0 0 0 0 0 0 0 0 0 0 ...
$ InvestmentFromFriendsAmount        : num  0 0 0 0 0 0 0 0 0 0 ...
$ Investors                          : int  258 1 41 158 20 1 1 1 1 1 ...
```

```
ntheme = theme_grey()+theme(text = element_text(family = "CMU Serif",size = 24))
theme_set(ntheme)
```

### Matrix Of Plots

We make a couple of preliminary matrix plots to get a feel for the data. It would be nice to edit the variable names in the ggpairs command for easy readability. It was easier to just create another dataframe and rename the variables to a smaller name as a easy fix.

```
library(plyr)
# Create New DF with Shorter feature names
# helps in the labeling of the matrix plots
fd <- rename(
pd, c(
'CreditGrade' = 'CGrade', 'LoanStatus' = 'LStatus',
'BorrowerAPR' = 'bApr', 'LenderYield' = 'LYield',
'EstimatedEffectiveYield = EEYield', 'EstimatedLoss' = 'ELoss',
'EstimatedReturn' = 'EReturn', 'ProsperRating..numeric.' = 'Rating',
'ProsperScore' = 'PScore','BorrowerState' = 'BState','Occupation' =
'Occup',
'EmploymentStatus' = 'Estatus','CreditScoreRangeLower' = 'CSLow',
'CreditScoreRangeUpper' = 'CSUp','CurrentCreditLines' = 'CCLine',
'OpenCreditLines' = 'OCLine','TotalCreditLinespast7years' = 'TC7yrs',
'OpenRevolvingMonthlyPayment' = 'ORMpay', 'CurrentDelinquencies' = 'CrntDel',
'AmountDelinquent' = 'AmDel','DelinquenciesLast7Years' = 'Del7yrs',
'DebtToIncomeRatio' = 'D2IRat', 'LoanMonthsSinceOrigination' = 'LMnthOri',
'MonthlyLoanPayment' = 'MLPay',
'LP_CustomerPayments' = 'CusPay','LP_CustomerPrincipalPayments' = 'CusPPay',
'LP_GrossPrincipalLoss' = 'GrPrLs','LP_NetPrincipalLoss' = 'NtPrLs'
)
)
# matrix of plots
set.seed(10000)
# pdn (below) holds a fraction of the original DataFrame pd
pdn <- fd[sample(1:length(pd$LenderYield),10000),c(4,8,10,14,16,17,18,21,29,30,31,32,
                                                   33,37,38)]
p1 <- ggpairs(pdn,params = c(shape = I('.'),outlier.shape = I('.')))

suppressMessages(print(p1))
```

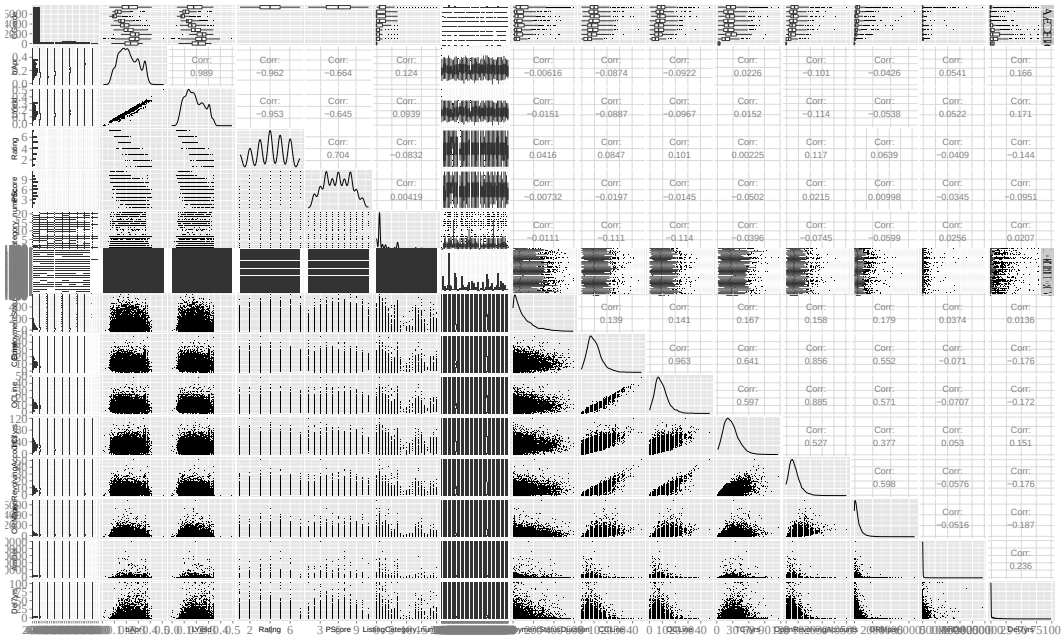

Figure 1: A matrixplot of variables from the first half of the features. Some interesting correlations are observed. See plot[3,2], i.e. EstimatedEffectiveYield vs BorrowerAPR, plot[(10,11,12,13),9] [(11,12,13),10]. We might compare some of these variables (BorrowerAPR, EstimatedEffectiveYield, OpenCreditLines, TotalCreditLinespast7years,OpenRevolvingMonthlyPayment) with variables from the next plot.

```
pdn = fd[sample(1:length(pd$LenderYield),10000),c(38,39,47,48,50,54,57,58,60,62,68,69,
                                                  70,75)]
p2 <- ggpairs(pdn,params = c(shape = I('.'),outlier.shape = I('.')))
suppressMessages(print(p2))
```

```
pdn  = fd[sample(1:length(pd$LenderYield),10000),c(48,58,60,62,68,69,70,75)]
p3 <- ggpairs(pdn,params = c(shape = I('.'),outlier.shape = I('.')))
suppressMessages(print(p3))
```

```
pdn = fd[sample(1:length(pd$LenderYield),10000),c(4,8,11,12,14,16,20)]
p4 <- ggpairs(pdn,params = c(shape = I('.'),outlier.shape = I('.')))
suppressMessages(print(p4))
```

```
pdn = fd[sample(1:length(pd$LenderYield),10000),c(4,8,11,12,14,16,20,31, 38,58,62,75)]
p5 <- ggpairs(pdn,params = c(shape = I('.'),outlier.shape = I('.')))
suppressMessages(print(p5))
```

```
# ProsperRating as a factor
pd$ProsperRating..numeric. = as.factor(pd$ProsperRating..numeric.)
# Loan TermPeriod as a factor
pd$Term = as.factor(pd$Term)
# Order the Grades in decreasing order
pd$CreditGrade = ordered(pd$CreditGrade,c("AA", "A","B", "C", "D", "E", "HR","NC",""))
```

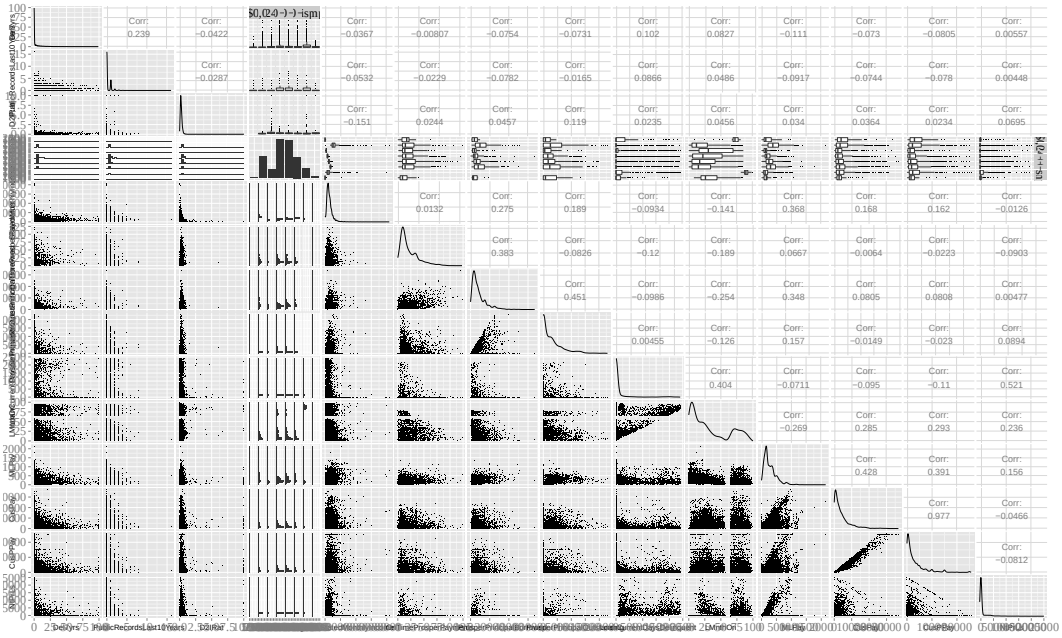

Figure 2: A scatterplot matrix of variables from the second half of the feature space. We see some negative correlation in the loss vs customer Principal Payments and loss vs Customer Payments. Also notice some positive correlation in MonthlyLoanPayment and above variables.

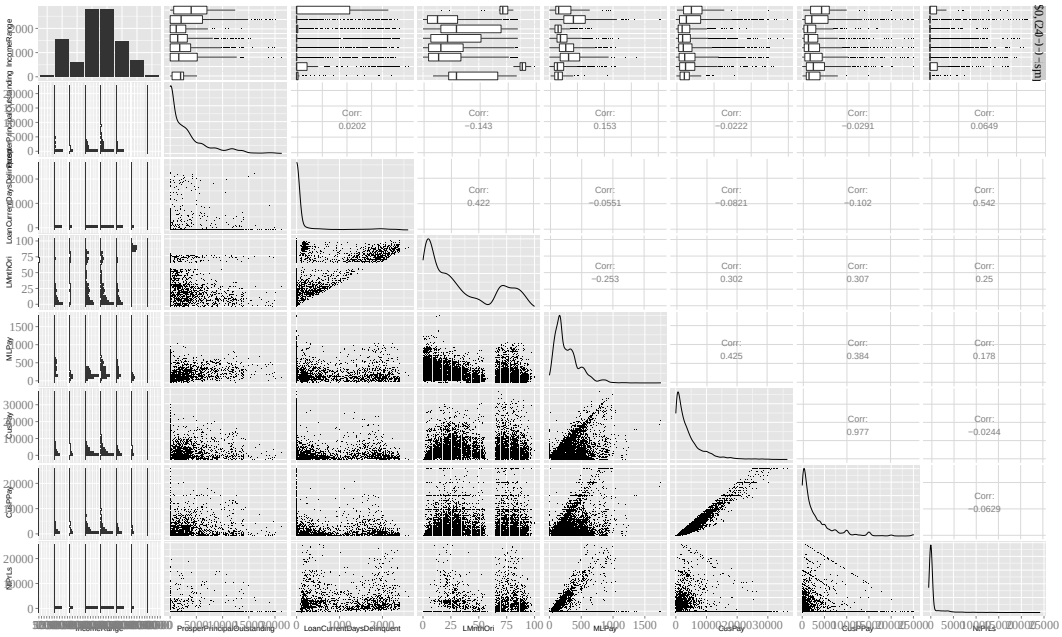

Figure 3: Notice negative correlations at the bottom right of the matrix. In the LoanMonthsOriginal column we see that around the 60 month mark there is a break. This needs to be looked in to more carefully. Notice positive correlations between MonthlyLoanPayment and NetLoss and LP_CustomerPayments and LP_CustomerPrincipalPayments

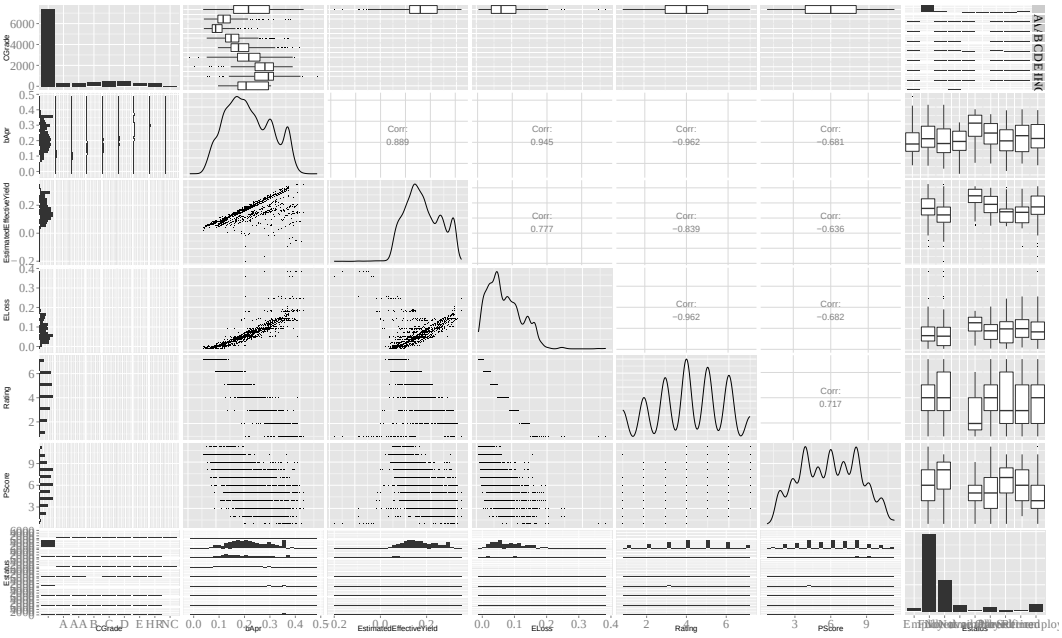

Figure 4: Variables like EstimatedLoss, EstimatedEffectiveYield, ProsperRating..numeric., and PropserScore when shown against CreditGrade have only one column (see top row of matrix plot) indicating the lack of CreditGrade information for those variables. Reading the description we see that CreditGrade is only for loans prior to 2009. We can infer that the above mentioned variables are only for loans made later than 2009. Whereas BorroweAPR is present for all columns of CreditGrade and shows correlation with above variables indicating that it is a feature for all loans.

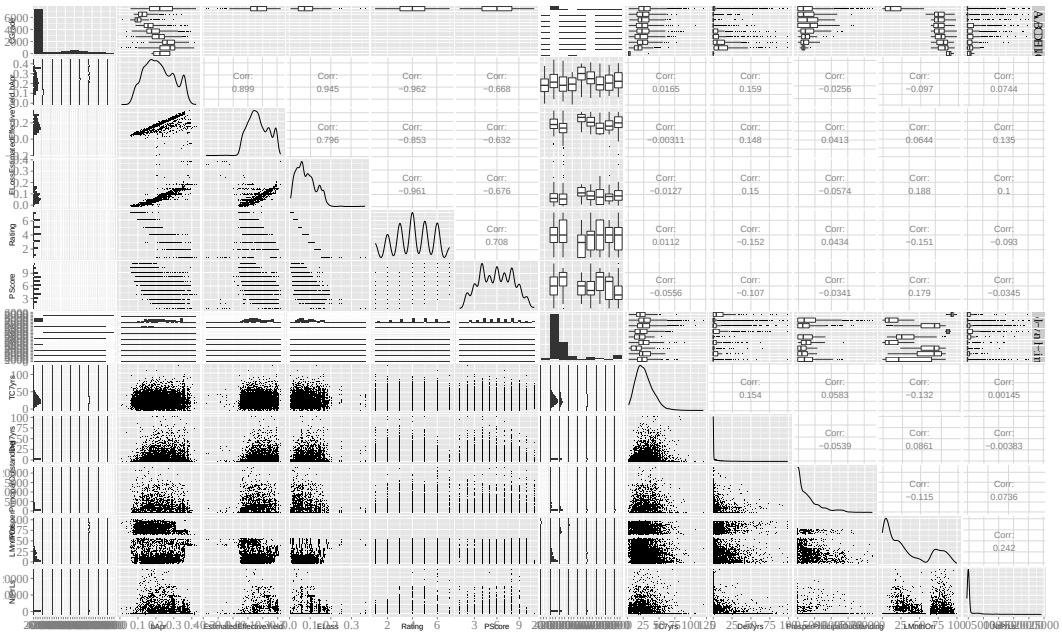

Figure 5: We see that some of the variables that should be factors are treated as numeric variables by noticing that we get a density plot instead of a histogram (e.g. ProsperRating..numeric.). Accordingly, we convert them to factor variables in the following section

```
# Order the ProsperScore
unique(pd$ProsperScore)
```

```
 [1] NA  7  9  4 10  2 11  8  5  3  6  1
```

```
pd$ProsperScore = ordered(pd$ProsperScore,c(1,2,3,4,5,6,7,8,9,10,11,NA))
```

# Exploratory Analysis

## Amount Delinquent

```
pds <-
  subset(pd,LoanCurrentDaysDelinquent > 0) # subset of delinquent loans
  qplot(
  data = pd,x = pd$LoanCurrentDaysDelinquent,binwidth = 50
  ) +
  scale_x_continuous(breaks = seq(0,3000,200),limits = c(1,3000)) +
  xlab('Days Delinquent')
```

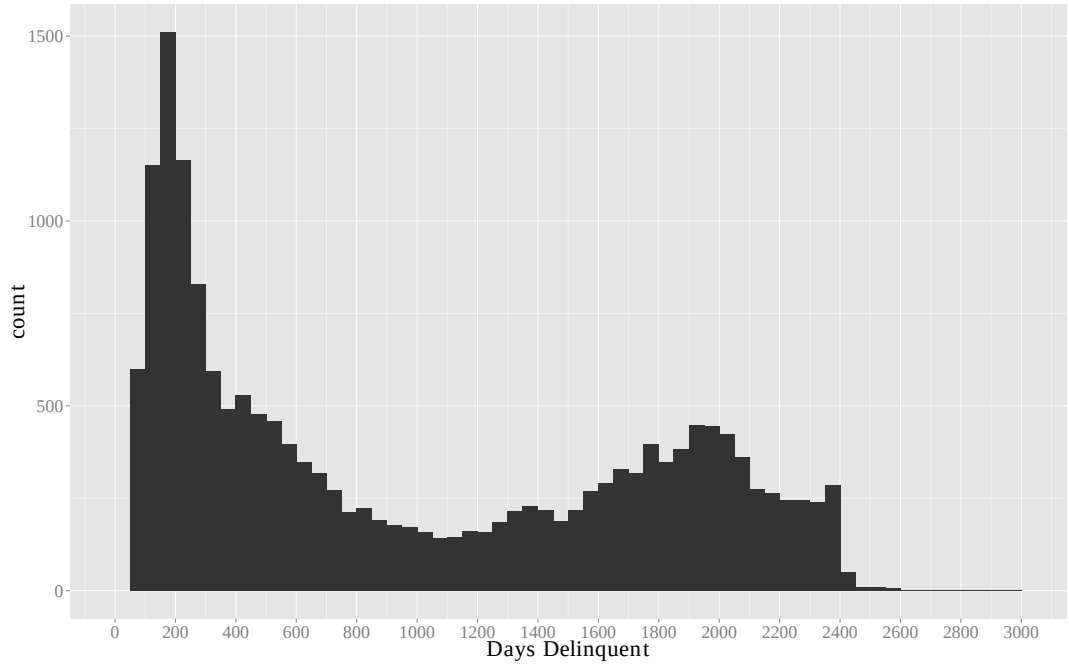


Figure 6: A histogram of the number of days loans have been delinquent. Looks like the number of delinquencies drop around the 3 year mark (1200 days) and then tend to increase towards the 2000 days mark. Very few loans are delinquent past the 2500 days mark.

```
# Notice some people who are delinquent for more than 2500 days!!
# The number of people who remain delinquent drops after 1000 days. It further incrases
#to 2000 days and just before 2500 days there is an abrupt drop in the number of
#delinquent loans indicating that the loans were probably charged off after that limit.
#There are a few "extraodrindary" loans that survive the 2500 day mark.
```

Consider the variation of delinquent amount and the number of delinquencies the borrower has had in the last 7 years.

```
    npd <- pd[pd$AmountDelinquent > 0 & !is.na(pd$AmountDelinquent &
                                                pd$LoanCurrentDaysDelinquent > 0.1),]
    npd <- npd[npd$LoanCurrentDaysDelinquent > 0,]
    ggplot(data = npd,aes(x = LoanCurrentDaysDelinquent,y = AmountDelinquent)) +
    geom_point( size = 1,position = 'jitter',alpha = 1,color = 'red') +
    geom_line(linetype = 1,stat = "summary",fun.y = median) +
    xlab('Current Days Delinquent') +
    ylab('AmountDelinquent') + ylim(c(0,1.5e5))
```

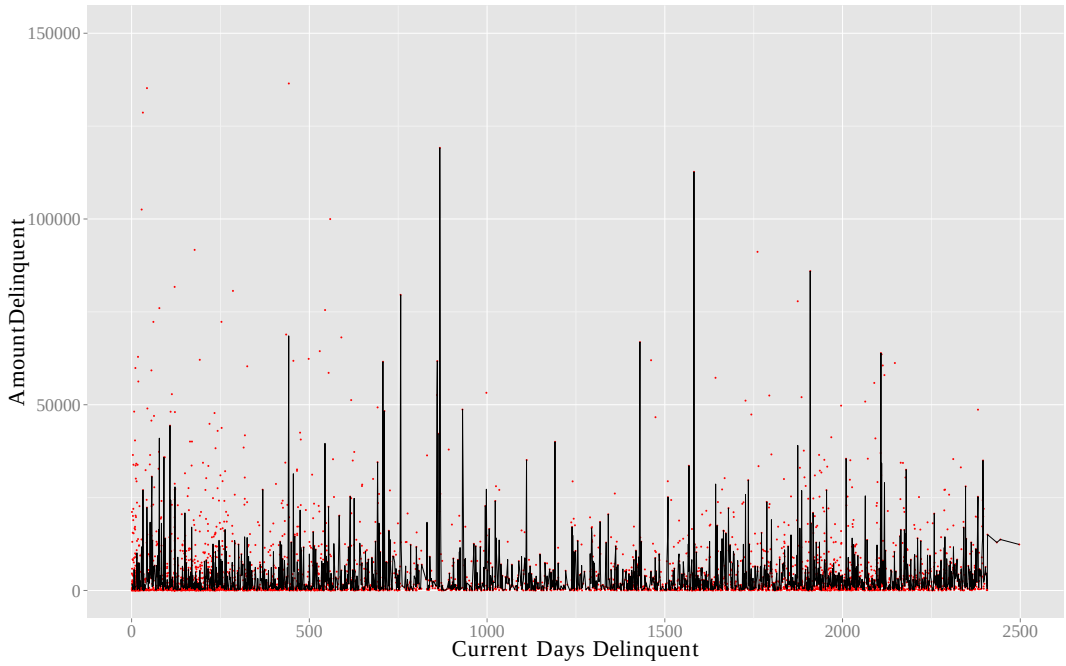


Figure 7: No particular relation is discernible by plotting the variables in regular scale. We therefore switch to log scale to look at the relationship between these variables.

```
    geom_smooth(method = "lm")
```

```
geom_smooth:
stat_smooth: method = lm
position_identity: (width = NULL, height = NULL)
```

```
    # It is hard to see any relation between the variables.
```

Looking at the same variables in log-scale reveals a hidden relationship.

```
npd <- pd[pd$AmountDelinquent>0 & !is.na(pd$AmountDelinquent
                                        & pd$DelinquenciesLast7Years>0.1),]
npd <- npd[npd$DelinquenciesLast7Years>0,]
pamtDelinq <- ggplot(data=npd,aes(x=DelinquenciesLast7Years,y=AmountDelinquent))+
  geom_point(size=1,position = 'jitter',alpha=1/5,color='red')+
```

```
  geom_line(stat='summary',fun.y=quantile,probs=0.9,linetype=2,color='blue')+
  geom_line(linetype=1,stat="summary",fun.y=median)+
  geom_line(stat='summary',fun.y=quantile,probs=0.1,linetype=2,color='blue')+
  scale_x_log10()+scale_y_log10()+
  stat_smooth(formula = y~x)+
  xlab('Log of Delinquencies in Last 7 years')+ylab('Log of AmountDelinquent')
print(pamtDelinq)
```

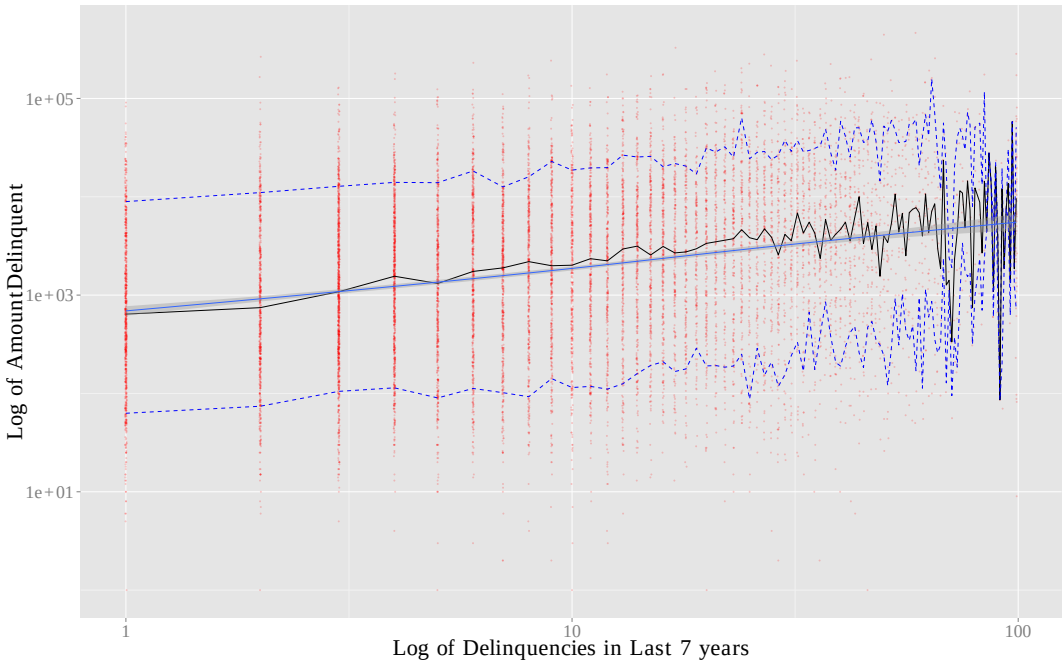


Figure 8: When we look at the variables on a log scale we see that there is a linear increase in the mean log delinquent amount with the number of delinquencies in the last 7 years. We see that there is a relationship of the form $y = kx^m$, where y = mean of AmountDelinqeint and $x$= mean of Current Days Delinquent.

```
#
```

## Credit Grade

Plot By Credit Grade Reorder the credit grades in the right order.

```
# Reorder the grades
pd$CreditGrade= ordered(pd$CreditGrade,c("AA", "A","B", "C", "D", "E", "HR","NC",""))
pd$ProsperRating..Alpha.= ordered(pd$ProsperRating..Alpha.,c("AA", "A","B", "C", "D", "E", "HR","",NA))
library(plyr)
pd$CreditGrade = mapvalues(pd$CreditGrade, from = "", to = "NV")
pd$pd$ProsperRating..Alpha. = mapvalues(pd$ProsperRating..Alpha., from = "", to = "NV")
# NV for "No Value""

# plot of Amount Delinquent vs. CreditGrade.
ggplot(aes(x=CreditGrade,y=AmountDelinquent),data=pd[pd$AmountDelinquent>0,])+
  geom_boxplot()+ylim(c(0,2.5e3))
```

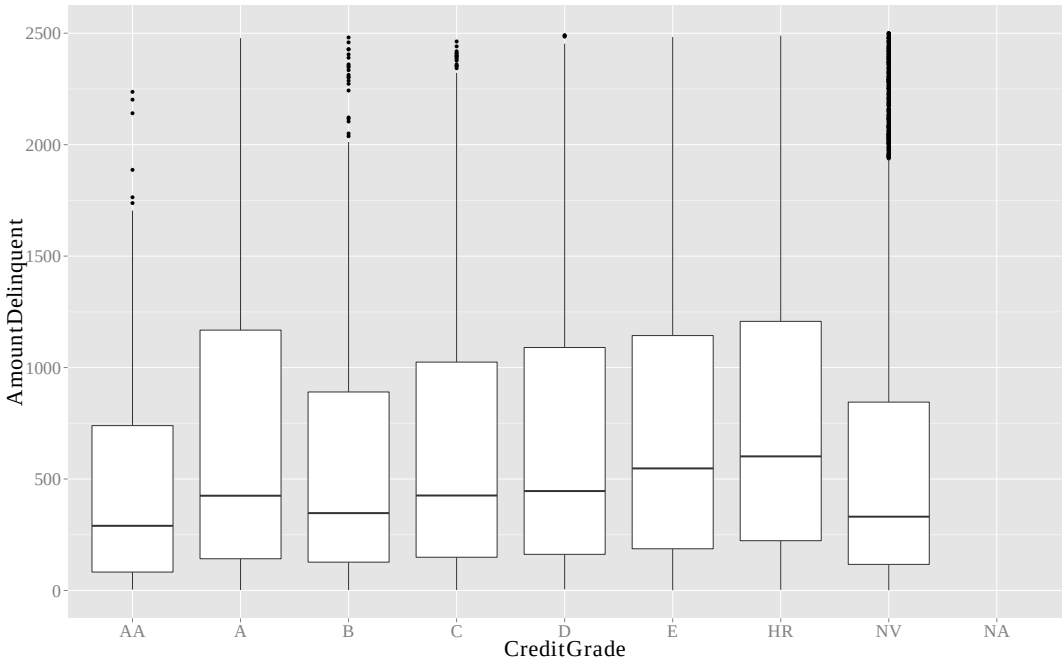


Figure 9: We check if the amount delinquent on loans is related to the credit rating of the borrower. We find that (with the exception of A and AA) the mean amount delinquent shows an increase with worsening credit grade.

```
ggplot(aes(x=ProsperRating..Alpha.,y=AmountDelinquent),data=pd[pd$AmountDelinquent>0,])+
  geom_boxplot()+ylim(c(0,2.5e3))+
  guides(color=guide_legend(title="Rating", override.aes =
                                          list(size=5)))
#See Fig. 10
```

```
qplot(x=pds$LP_CustomerPrincipalPayments/pds$LoanOriginalAmount,
      y=pds$AmountDelinquent/pds$LoanOriginalAmount,data=pds,color=ProsperScore)+
  scale_y_log10()+xlim(0,0.5)
#See Fig. 11
```

```
# Comparing the Borrower APR and Prosper Rating
ggplot(aes(x=ProsperRating..numeric.,y=BorrowerAPR),
       data=pd[!is.na(pd$ProsperRating..numeric.),])+
  geom_boxplot()+scale_x_discrete()+xlab("Prosper Rating")
#See Fig. 12
```

```
#See Fig. 12
# This figure shows that as a lender's rating worsens, the lender tries to mitigate his
# risk by levying a heavier interest rate on the loan. For future borrowers, just this
# plot should be incentive to maintain a high credit rating.
```

```
ggplot(aes(y=LP_NetPrincipalLoss/LoanOriginalAmount,x=CreditGrade),data=pd[pd$LP_NetPrincipalLoss>0,],b
geom_boxplot()+ylab('Net Prin. Loss/ Loan Orig. Amt')
#See Fig. 13
```

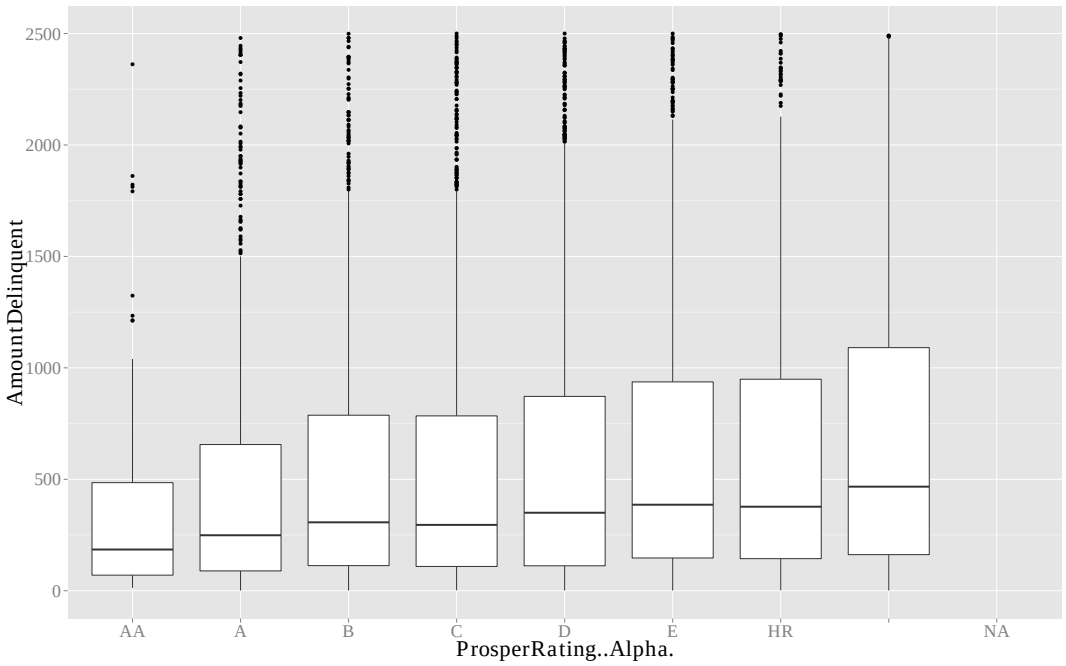


Figure 10: By plotting the AmountDelinquent vs. Credit grade determined by prosper the relation pointed out in the previous plot is more regular (without the exception of A and AA credit grades, as in previous figure). The mean delinquent amount increases with worsening credit grade.

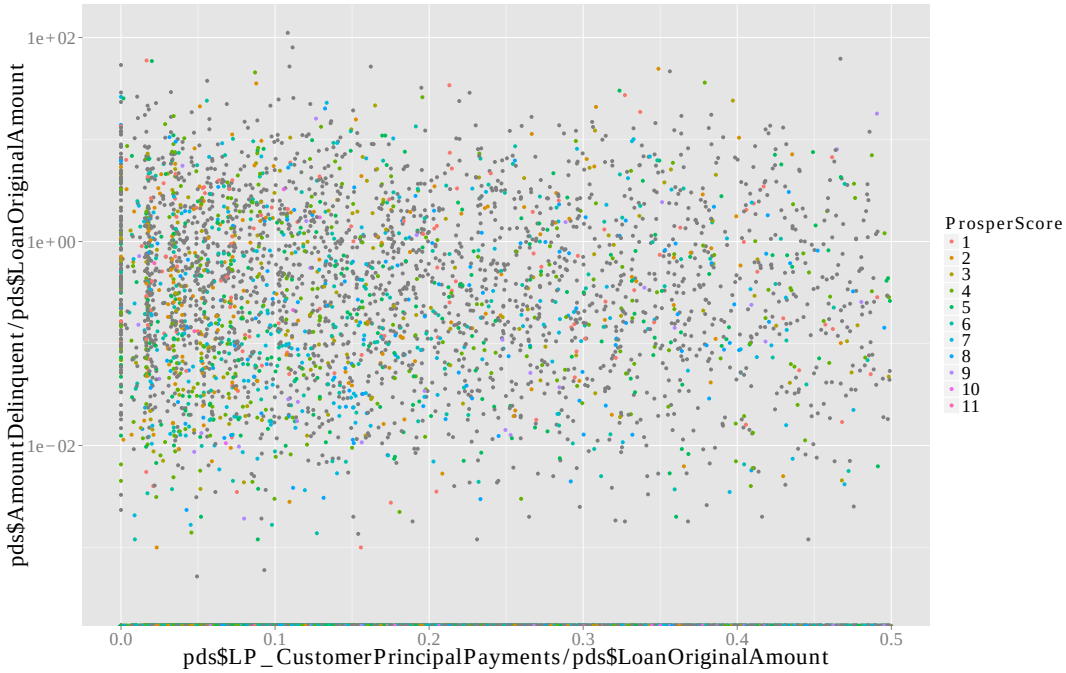


Figure 11: We show the ratio of Delinquent Amount Vs Customer Principal Payments —both quantities being normalized by Loan Original Amount. There is no apparent relationship between the two variables and the coloring by ProsperScore does not indicate any trend.

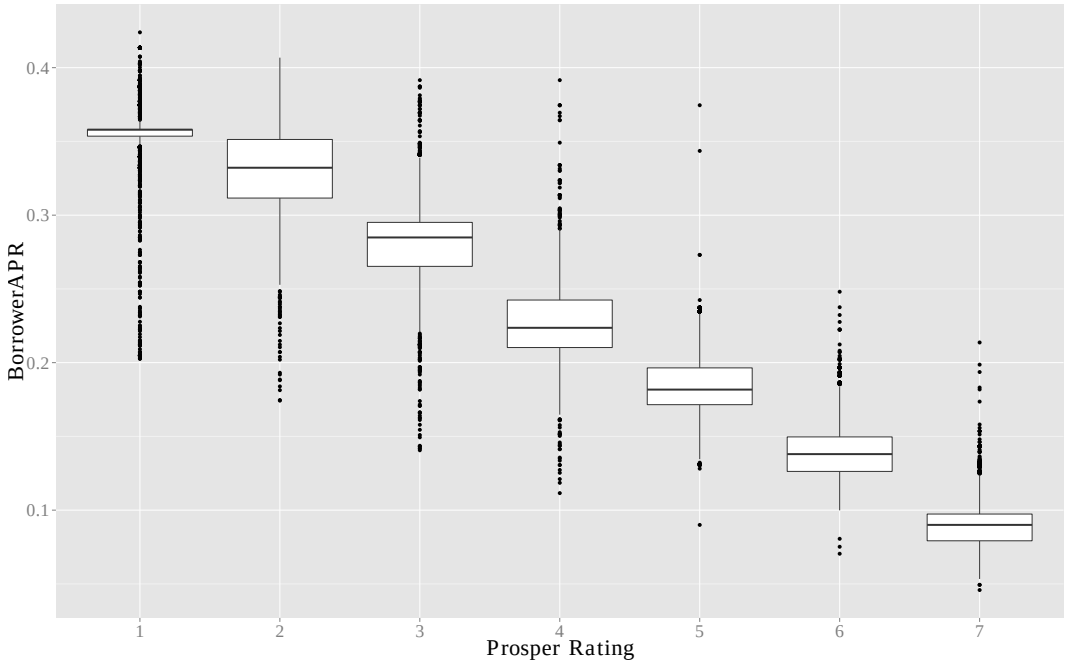


Figure 12: A very clean decrease of mean BorrowerAPR is found with decreasing ProsperRating. Note that higher numbers correspond to better rating with 7 = AA and 1= HR (high risk).

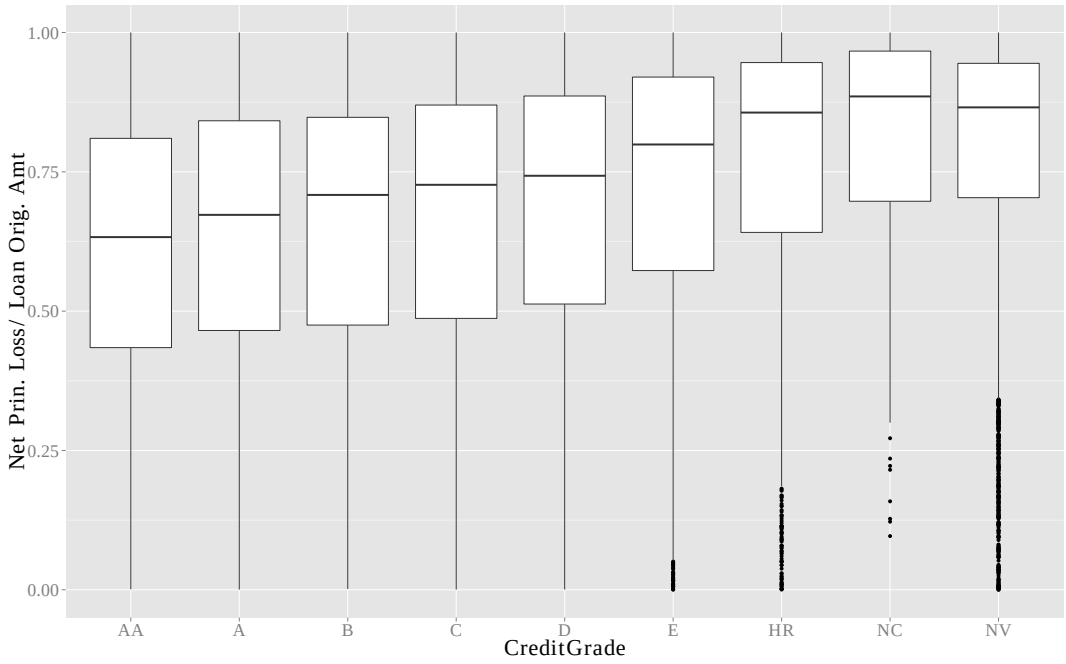


Figure 13: The LP_NetPrincipalLoss (normalized by Loan Amount) increase as the credit grade worsens.

**By State**

```
# Normalize AmountDelinquent by LoanOriginalAmount - plot in regular coordinates
pds <- pd[pd$AmountDelinquent>0 & !is.na(pd$LoanOriginalAmount) & !is.na(pd$LoanOriginalAmount)>0,]
ggplot(aes(x=BorrowerState,y=AmountDelinquent/LoanOriginalAmount),data=pds)+
  geom_boxplot()+ylab('Delinquent Amount / Original Loan')+
  xlab('Borrower State')+
  theme(axis.text.x =element_text(angle=0,size=10))+coord_cartesian(ylim=c(0,2.5))
#See Fig. 14
```

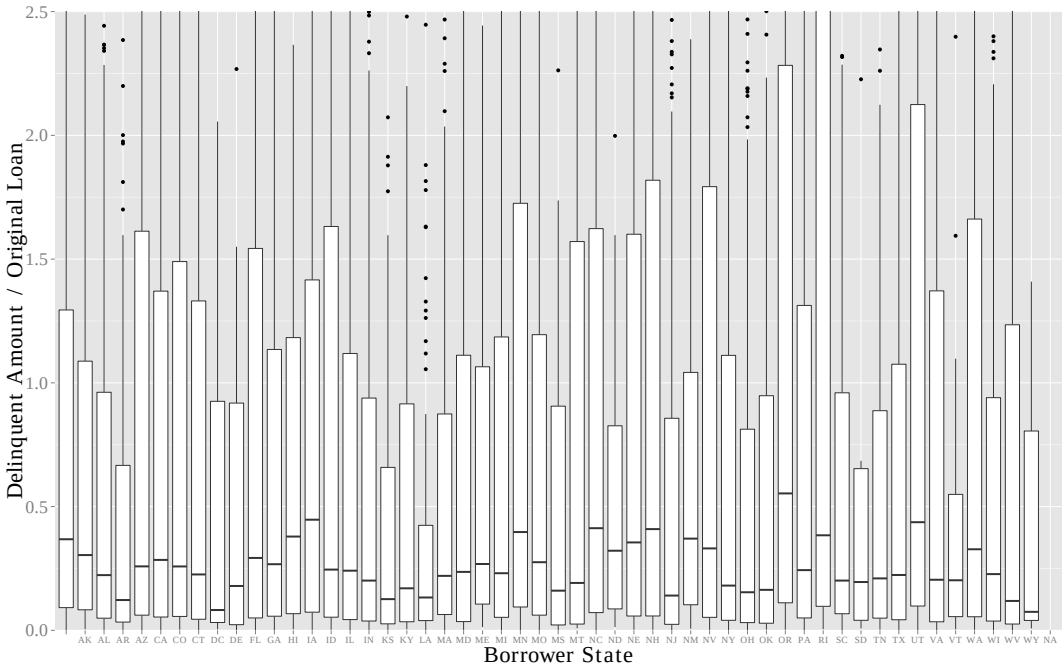


Figure 14: The normalized Delinquent Amount (normalized by Original Loan amount is shown for all the states). California has the most delinquent loans. Wyoming has the fewest loans. To zoom the y-axis we have to be sure that we don't omit data and therefore use the coord_cartesian(ylim=limits) option.

```
# Normalize AmountDelinquent by LoanOriginalAmount - plot in log-scale
# Compare with previous plot log vs. non-log scales)
ggplot(aes(x=BorrowerState,y=AmountDelinquent/LoanOriginalAmount),data=pds)+
  geom_boxplot()+scale_y_log10()+ylab('Delinquent Amount / Original Loan')+
  xlab('Borrower State')+
  theme(axis.text.x =element_text(angle=0,size=10))
# From the Log-scale we see that sometimes the Amount Delinquent is more than
10 times greater than Original Loan.
#See Fig. 15
```

```
# Piping commands in dplyr to find state with Max & Min of the mean Normalized Amount Delinquent
library(dplyr)
pdAmtDelinqRatio<- pd[pd$AmountDelinquent>0 & !is.na(pd$AmountDelinquent),] %>%
                          group_by(BorrowerState) %>%
                          dplyr::summarise(mean_AmtDelinqRatio= mean(AmountDelinquent/LoanOriginalAmount
```

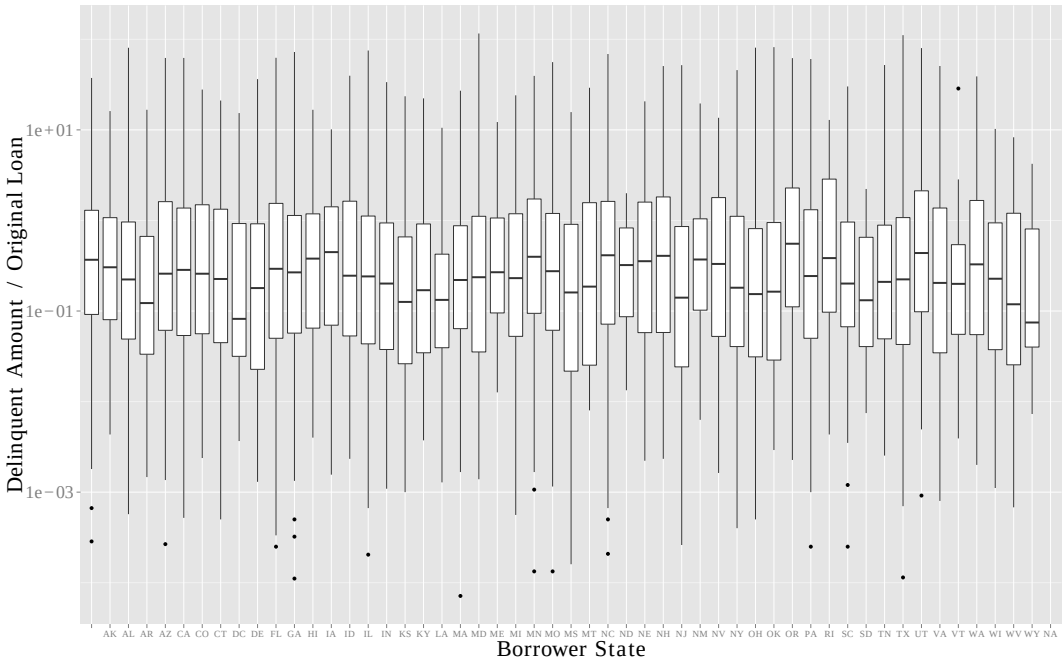


Figure 15: We look at the same plot as above but in the log-scale hoping to see better resolution. Unfortunately this does not clarify any new information than plotting it in regular cartesian scale. The log-scale however enables us to see that in some extreme cases the delinquent amount is almost 10 times the original loan amount.

```
                          n=n()) %>%
                          arrange(BorrowerState)
pdAmtDelinqRatio[pdAmtDelinqRatio$n== max(pdAmtDelinqRatio$n),]
```

```
Source: local data frame [1 x 3]

  BorrowerState mean_AmtDelinqRatio     n
         (fctr)               (dbl) (int)
1           CA            1.806464  2021
```

```
pdAmtDelinqRatio[pdAmtDelinqRatio$n== min(pdAmtDelinqRatio$n),]
```

```
Source: local data frame [1 x 3]

  BorrowerState mean_AmtDelinqRatio     n
         (fctr)               (dbl) (int)
1           WY           0.7722202     9
```

```
# Group By State
pdAmtDelinqByState <- pd[pd$AmountDelinquent>0 & !is.na(pd$AmountDelinquent),] %>%
                          group_by(BorrowerState) %>%
                          dplyr::summarise(mean_Amount_Delinq = mean(AmountDelinquent),
                          n=n()) %>%
                          arrange(BorrowerState)
summary(pdAmtDelinqByState)
```

```
 BorrowerState mean_Amount_Delinq       n
        : 1    Min.   : 1618      Min.   :   9.0
 AK     : 1    1st Qu.: 3965      1st Qu.:  65.5
 AL     : 1    Median : 5458      Median : 190.0
 AR     : 1    Mean   : 5906      Mean   : 317.2
 AZ     : 1    3rd Qu.: 6704      3rd Qu.: 428.2
 CA     : 1    Max.   :18499      Max.   :2021.0
 (Other):46
```

```
# Plot by State
ggplot(aes(x=BorrowerState,y=mean_Amount_Delinq),data=pdAmtDelinqByState)+
geom_bar(stat="identity")+theme(axis.text.x =element_text(angle=0,size=10))+
ylab('Mean Amount Delinquent')
```

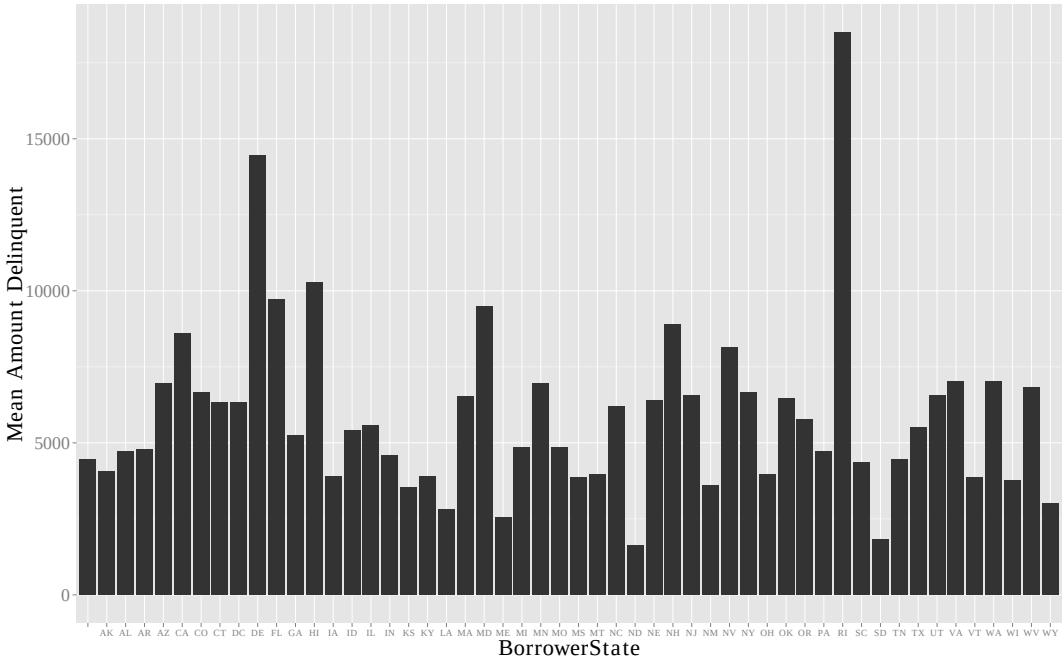


Figure 16: This figure shows the statewise mean amount delinquent. We could run similar stats as before and figure out which states have the highest and lowest Mean Delinquent amount. This could be normalized by the original loan amounts as before.

```
# Max and Min Absolute Delinquencies
pdAmtDelinqByState[pdAmtDelinqByState$n== max(pdAmtDelinqByState$n),]
```

```
Source: local data frame [1 x 3]

  BorrowerState mean_Amount_Delinq     n
         (fctr)              (dbl) (int)
1            CA           8593.425  2021
```

```
pdAmtDelinqByState[pdAmtDelinqByState$n== min(pdAmtDelinqByState$n),]
```

```
Source: local data frame [1 x 3]
```

```
  BorrowerState mean_Amount_Delinq     n
         (fctr)              (dbl) (int)
1            WY           2998.667     9
```

### Yield, Returns, and Other Plots

```
# We could do similar analysis for Yield by Credit Rating.
# Credit Grade is available for loans before 2009 and Prosper Rating for loans after 2009.
before2009 = pd[as.POSIXct(pd$LoanOriginationDate) < as.POSIXct('2009-01-01 00:00:00'),
                'CreditGrade']
after2009 = pd[as.POSIXct(pd$LoanOriginationDate) >=as.POSIXct('2009-01-01 00:00:00'),
               'ProsperRating..Alpha.']

# Create new variable called ConsolCreditGrade so that credit grade of all loans can be accessed
pd[as.POSIXct(pd$LoanOriginationDate) < as.POSIXct('2009-01-01 00:00:00'),
   'ConsolCreditGrade'] <- before2009
pd[as.POSIXct(pd$LoanOriginationDate) >= as.POSIXct('2009-01-01 00:00:00'),
   'ConsolCreditGrade'] <- after2009
```

```
# Convert ConsolCreditGrade to an ordered variable
pd$ConsolCreditGrade= ordered(pd$ConsolCreditGrade,
                levels=c("AA", "A","B", "C", "D", "E", "HR","NC","NV",NA))

spd <- pd[!is.na(pd$EstimatedLoss) & !is.na(pd$EstimatedEffectiveYield),]
pdLossYield <-  pd %>%
                group_by(ConsolCreditGrade) %>%
                summarize(mean_Estimated_Loss = mean(EstimatedLoss,na.rm=TRUE),
                mean_Effective_Yield = mean(EstimatedEffectiveYield,na.rm=TRUE)) %>%
                ungroup()%>%
                arrange(ConsolCreditGrade)
```

```
qplot(x=ConsolCreditGrade,y=mean_Estimated_Loss,data=pdLossYield)+
  geom_bar(stat="identity")+xlab('Credit Grade')+ylab('Estimated Loss (Mean)')
#See Fig.17
```

```
qplot(x=ConsolCreditGrade,y=mean_Effective_Yield,data=pdLossYield)+
  geom_bar(stat="identity")+xlab('Credit Grade')+ylab('Effective Yield (Mean)')
#See Fig.18
```

## By State

```
# Statewise count of the number of loans in each category.
pdQ <-  pd %>%
                group_by(BorrowerState,ListingCategory..numeric.) %>%
                summarize(count = n(),
                          Amount = sum(LoanOriginalAmount)) %>%
                arrange(BorrowerState)
```

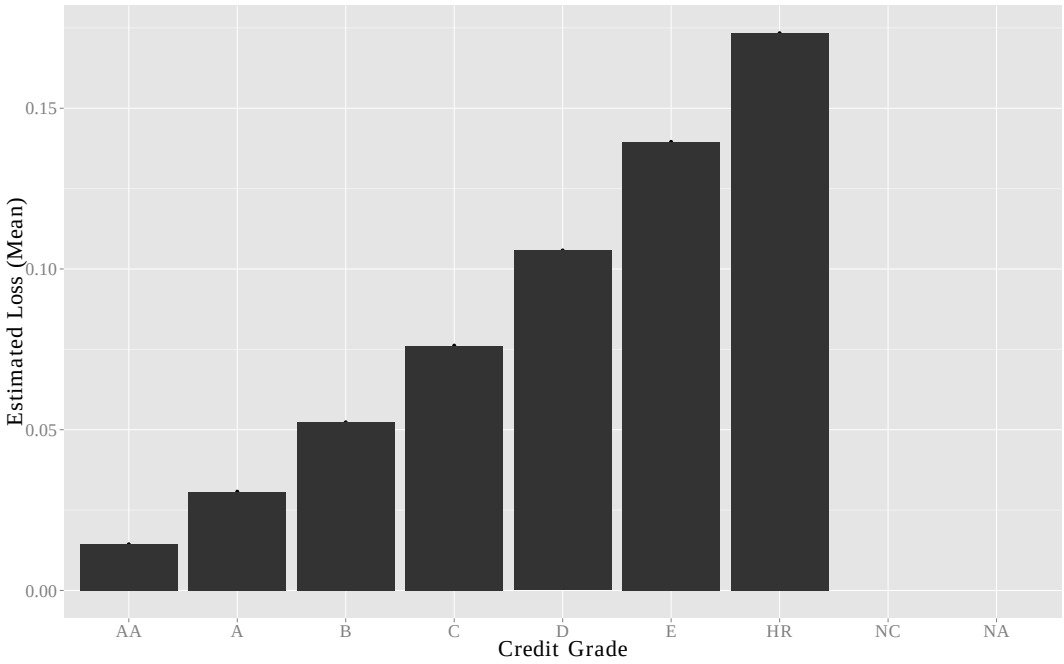


Figure 17: Plot of Mean Estimated Loss vs the Credit Grade. There is a clear linear trend showing the increase in Estimated Loss with worsening Credit Grade.

```
pdQmost <- pdQ[!(pdQ$BorrowerState==""),] %>%
          group_by(ListingCategory..numeric.) %>%
          summarize(TotalNumber = sum(count),
                    TotalLoan = sum(Amount))%>%
          arrange(TotalNumber)
qplot(data=pdQmost,x=ListingCategory..numeric.,y=TotalNumber)+geom_bar(stat="identity")+
  scale_y_log10()+xlab('Category')+ylab('Number Of Loans')
#See Fig.19
```

```
qplot(data=pdQmost,x=ListingCategory..numeric.,y=TotalLoan)+
  geom_bar(stat="identity")+scale_y_log10()+xlab('Category')+ylab('Loan Amounts')
#See Fig.20
```

### Service fee

The difference between the *Lender Yield* and the *Borrower rate* is the service fee. Plotting the histogram of the difference between the two quantities we find that mostly the service fee is about .01%.

```
pd$SerFee = pd$BorrowerRate-pd$LenderYield
ggplot(data=pd)+geom_histogram(aes(x=SerFee),binwidth=0.001)
#See Fig.21
```

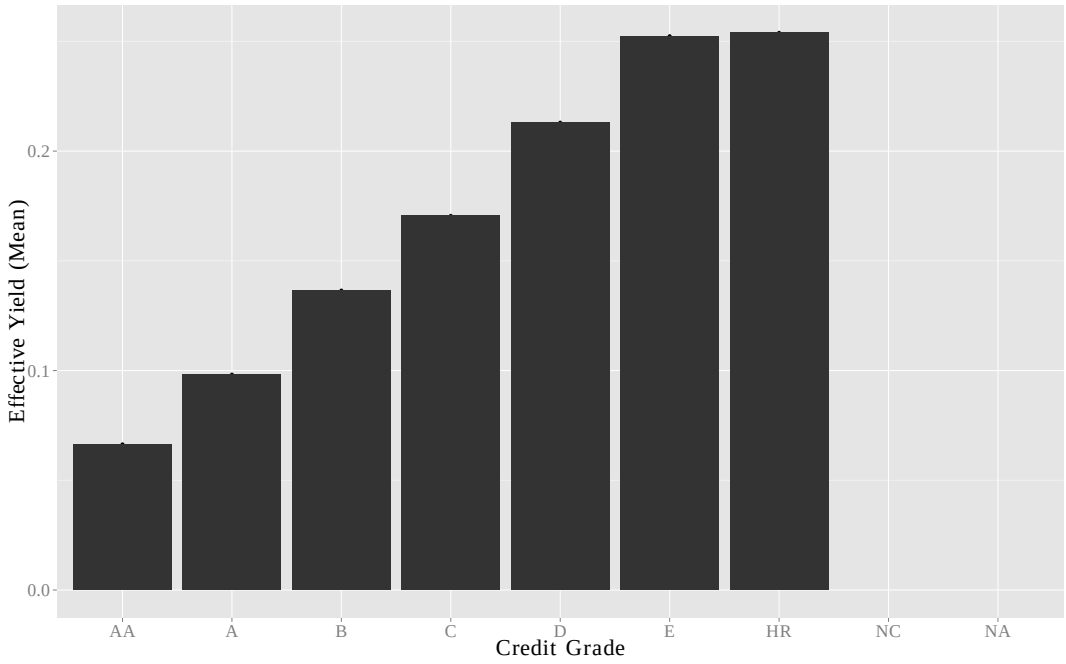


Figure 18: Plot of Mean Estimated EffectiveYield vs the Credit Grade. There is a clear linear trend showing the increase in EffectiveYield with worsening Credit Grade.

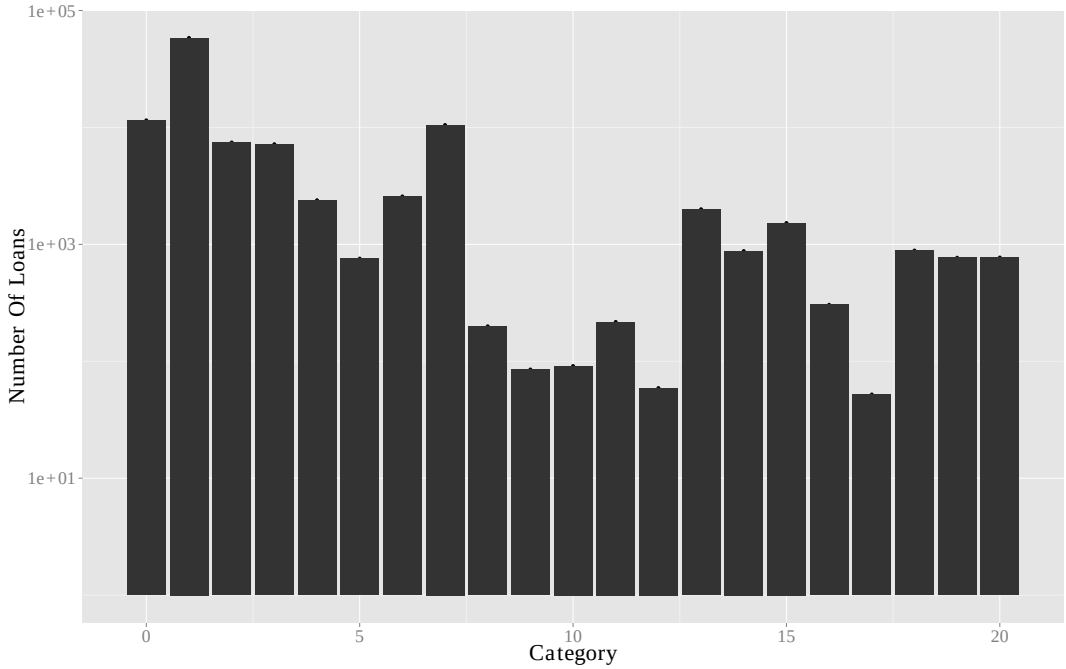


Figure 19: Number of Loans vs the category of Loan. We find that the "Debt Consolidation" is the most sought after loan type and motorcycle loans are the least.

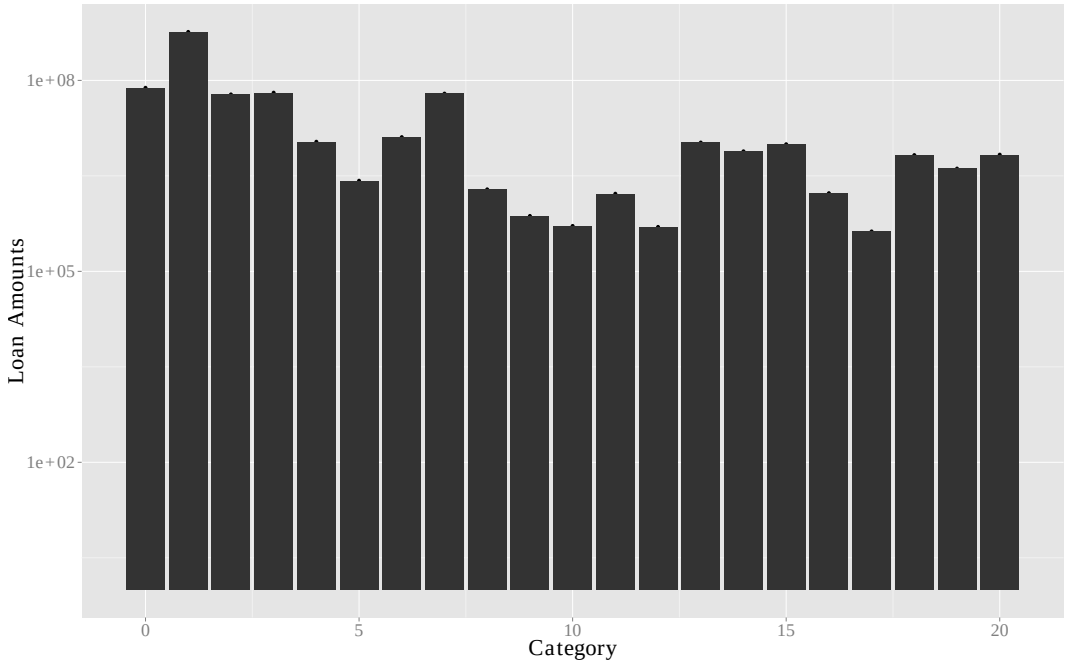


Figure 20: This is a similar plot as above. But instead of the number of loans made here we show the total amount of loans. Even in this category, "Debt Consolidation" tops the list and motorcycle loans are at the bottom.

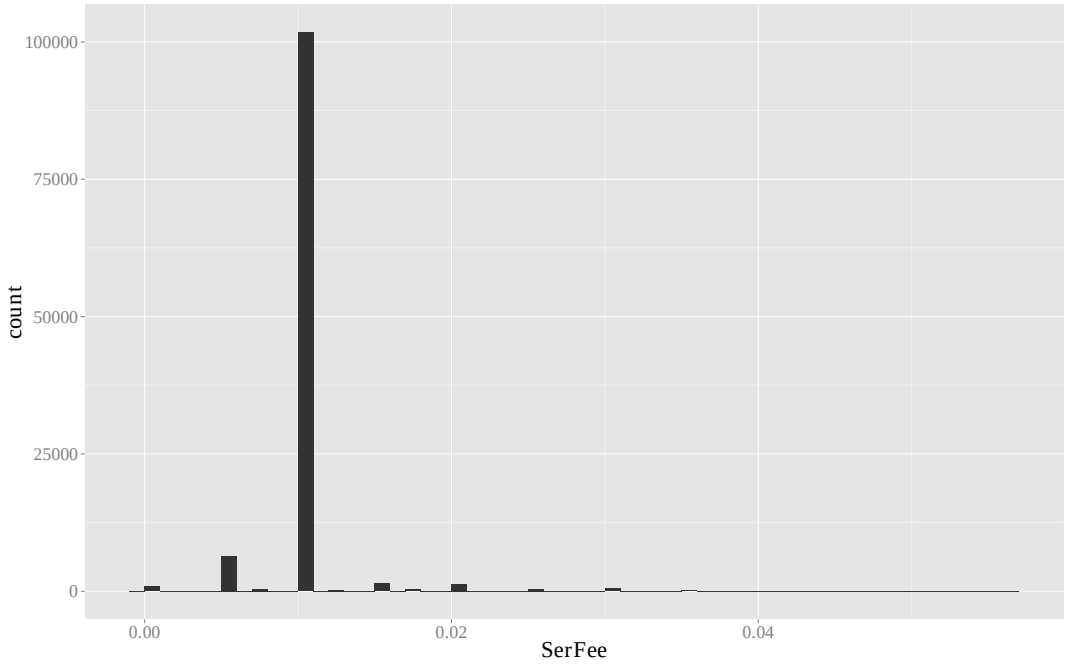


Figure 21: Here we show the histogram of the difference between BorrowerAPR and LenderYield. This difference is the Service Fee. We see that for most loans the Service fee is 0.01%.

## Loan Status and Loan Term

```
#Check if the Effective Yield is affected by the loan status
ggplot(aes(x = LoanStatus,y = EstimatedEffectiveYield),data = pd) + geom_boxplot() +
  theme(axis.text.x =  element_text(angle = 45,size = 15))
#See Fig.22
```

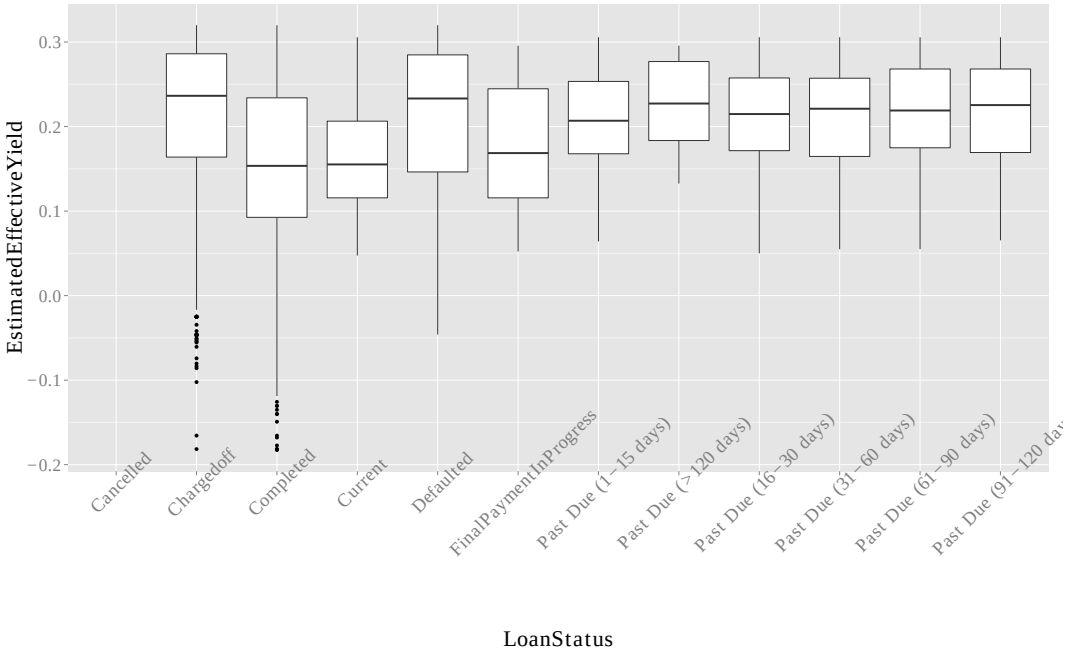


Figure 22: This plot shows the plot of the EffectiveYiled by Loan Status. If we do not consider the Charged Off and defaulted loans, there is a steady increase in Yield as the number of days past Due increases. This indicates that once a loan has gone beyond the due date, in addition to the interest on the loan, there is the late fees and other penalties that increase the yield.

```
# Reorder the factor LoanStatus.
pd$LoanStatus = ordered(pd$LoanStatus,c("Current","FinalPaymentInProgress","Completed","Past Due (1-15 d
# Replot the above plot with reordered Status
ggplot(aes(x = LoanStatus,y = EstimatedEffectiveYield),data = pd) + geom_boxplot() +
  theme(axis.text.x =  element_text(angle = 45,size = 20))
#See Fig.23
```

```
# How does Loan Status Influence Estimated Return
  ggplot(aes(x = LoanStatus,y = EstimatedReturn),data = pd) + geom_boxplot() +
  theme(axis.text.x =   element_text(angle = 45,size = 20))
#See Fig.24
```

```
  # We expect the Estimated Return to be correlated with the Lender Yield. How does the
  # Prosper Rating affect this relationship?
  ggplot(aes(x = LenderYield,y = EstimatedReturn,color = ProsperRating..numeric.),data = pd) +
  geom_point()+guides(color = guide_legend(title = "Rating", override.aes = list(alpha = 1,size = 5)))
#See Fig.25
```

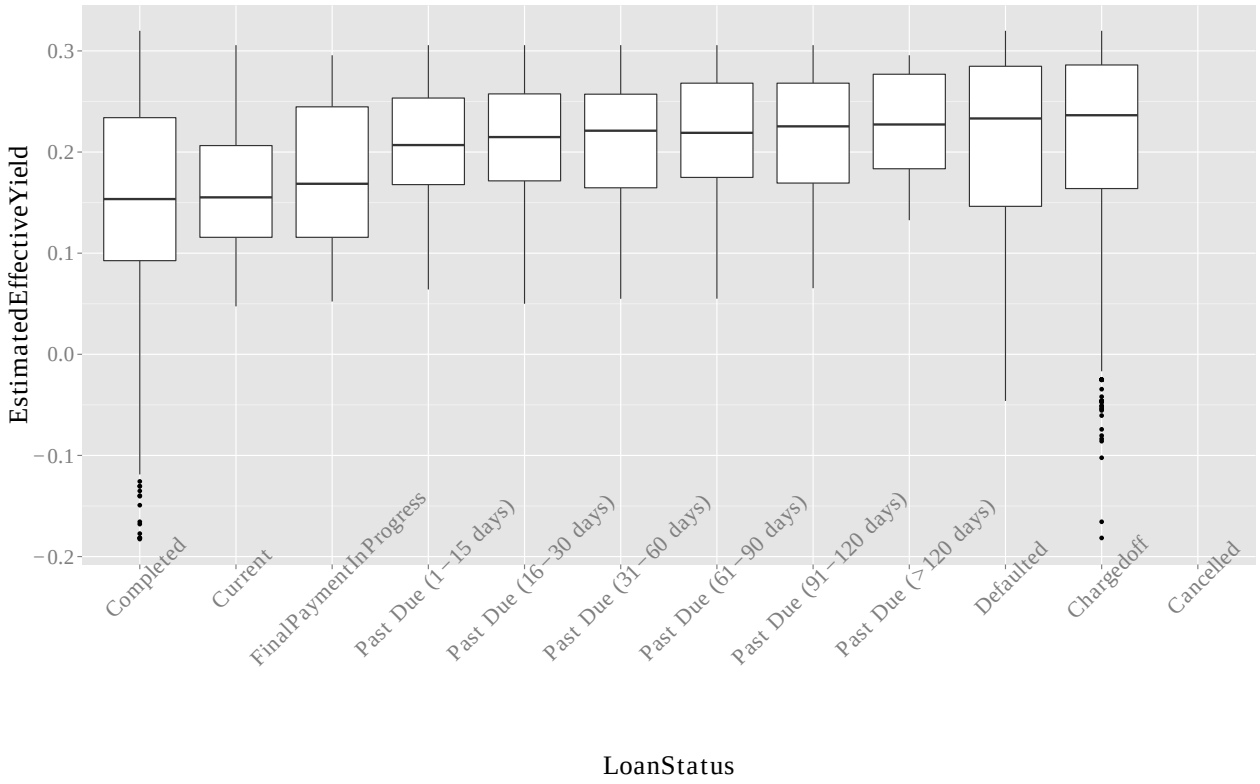


Figure 23: The same plot as above (i.e Effective Yield Vs. Loan Status) but with reordered Loan Status. There is a steady increase in the mean Yield as the number of days past dues increases and later defaults or is chargedoff. This indicates that the Lender is able to levy late fee and penalties to recuperate the uncollected amount.

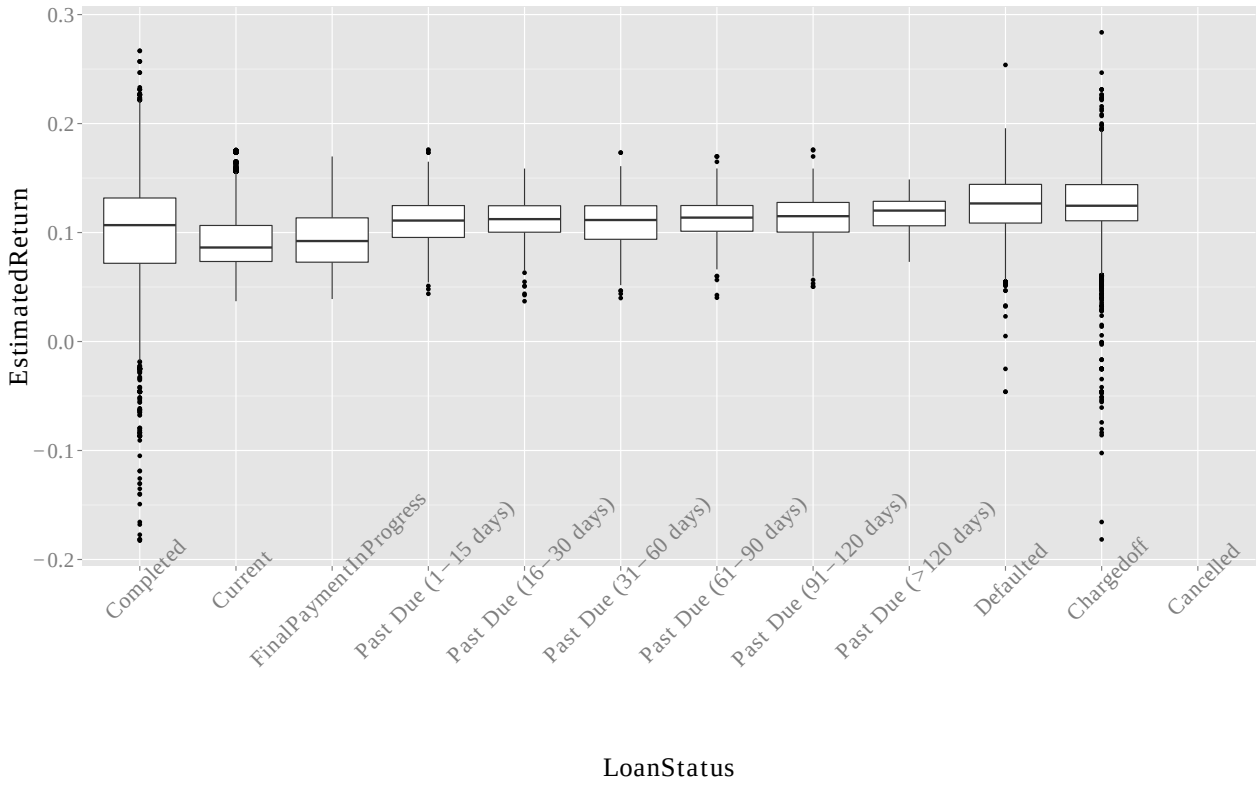


Figure 24: Estimated Return also shows a similar trend as the EstimatedYield. This means that the Lender makes more money (since Returns = Yiled - Loss) on the loans that show some delay or default.

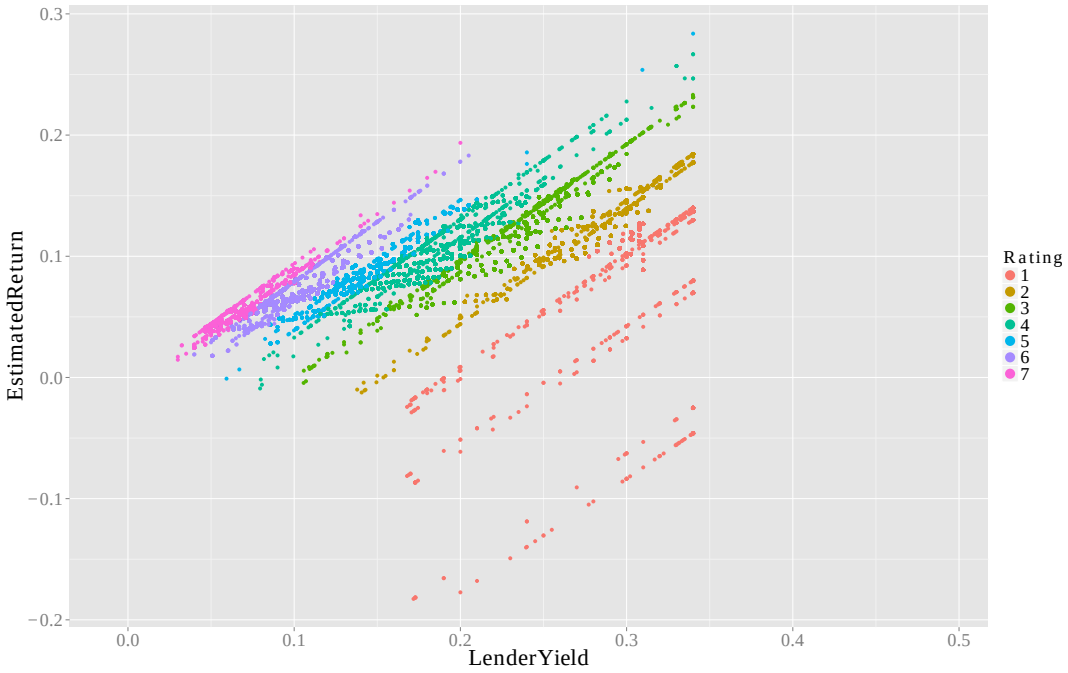


Figure 25: Plot of Return vs Yield. We see that there is a linear relationship between the two variables. On the top portion of the graph (i.e. EsimatedReturn $> 0.05$) we see that there are two regimes of linearities. Also for the same Prosper Rating (say Rating ==1) there are multiple linear zones of same slope. It is easy to observe that the slope is one and that indicates that the difference between Returns and LenderYield is a constant.

```
  #Notice that PrsperScore Offers better resolution to distinguish the bands with the
  # lowest Estimated Returns
  ggplot(data = pd,aes(x = LenderYield,y = EstimatedReturn,color = ProsperScore)) +
  geom_point()+guides(color = guide_legend(title = "Score", override.aes = list(alpha = 1,size = 5)))
#See Fig.26
```

```
  ### Instead of Prosper rating, using ProsperScore gives better distinction

  ggplot(aes(x = EstimatedEffectiveYield,y = EstimatedReturn),data = pd) +
  geom_point(color = 'purple') +
    guides(color = guide_legend(title = "Rating", override.aes = list(alpha = 1,size = 5)))
#See Fig.27
```

```
  # Loan Terms are 12, 36 and  60 months; separating by Loan Term
  # Term == 12 or 60
  ggplot(aes(x = EstimatedEffectiveYield,y = EstimatedReturn),data = pd[pd$Term !=36,]) +
  geom_point(color = 'blue')
#See Fig.28
```

```
  # Term == 36
  ggplot(aes(x = EstimatedEffectiveYield,y = EstimatedReturn),data = pd[pd$Term ==
  36,]) +
  geom_point(color = 'red')
#See Fig.29
```

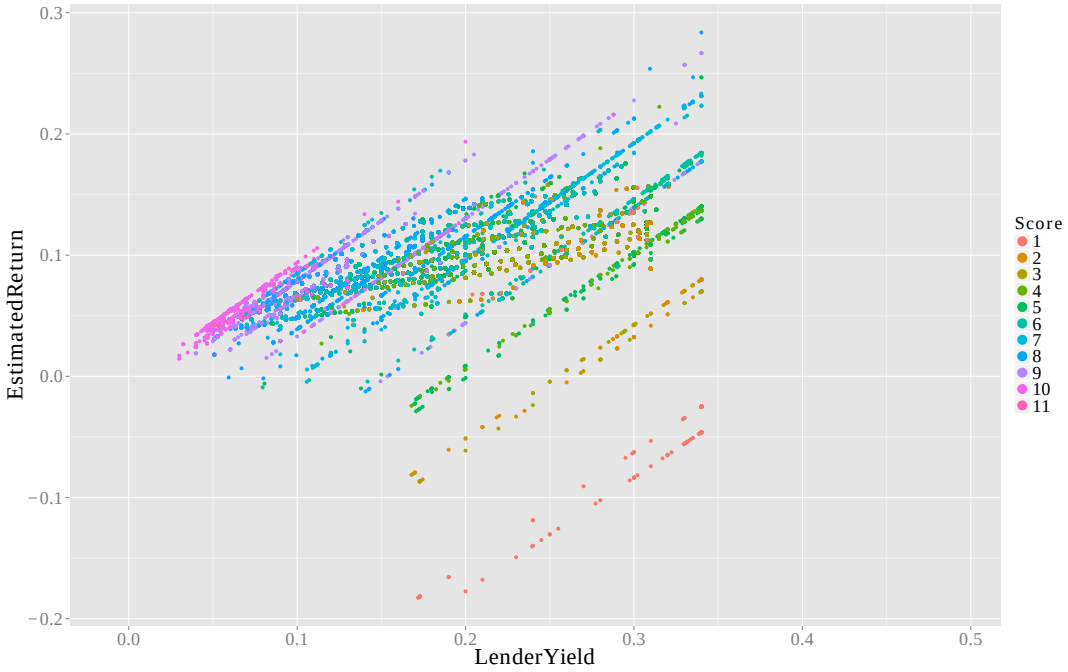


Figure 26: Same plot as above, but coloring using the ProsperScore provide better resolution on the linear zones even for the same Rating. We can surmise that the difference in the linear zones is because people with different Scores get different APRs and thus the returns are different based on the ProsperScores.

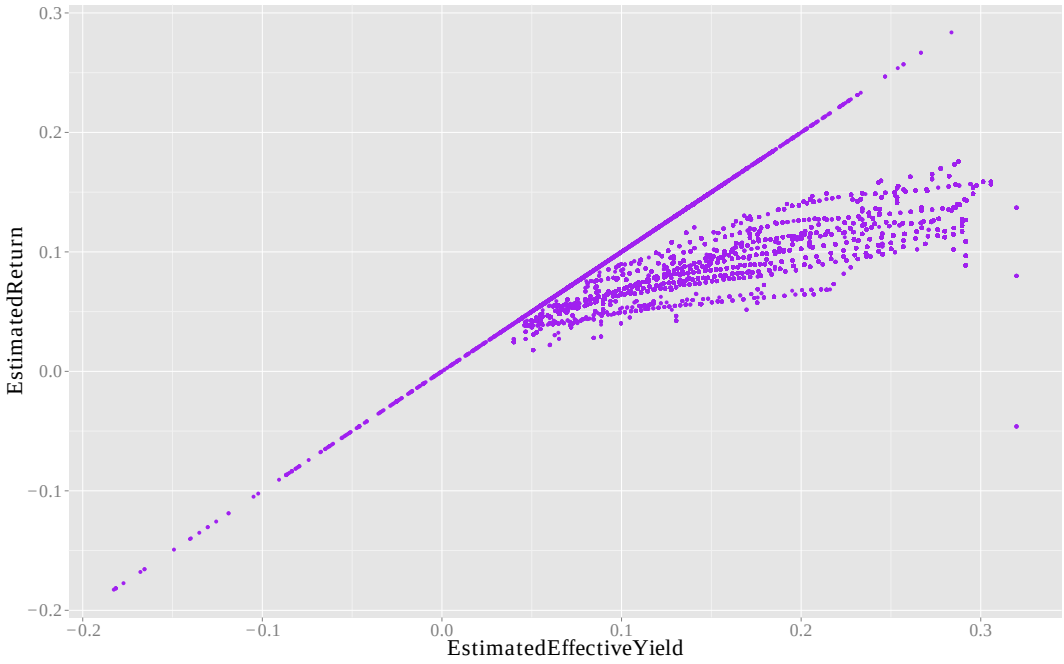


Figure 27: A plot of the Estimated Return vs the Effective Yield. Again, we see that there are different regimes of linear relationships. There is a whole subset of the data for whom the Return is identical to the Effective Yield. This is indicated by the long diagonal set of points that identify a line with slope = 1.

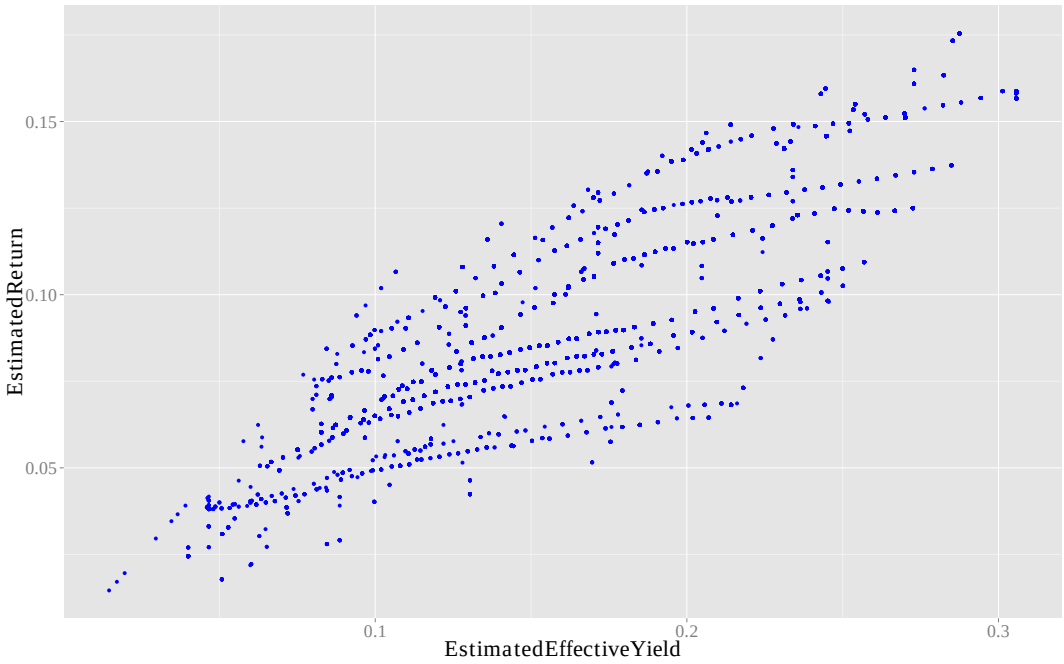


Figure 28: A plot of the EstimateReturn vs. Effective Yield for the loans with Term = 12 or 60 months. This separates the subset of the data where Return==yield for only those loans with Term=36 months.

```
  # There seem to be two regimes within the 36month Term itself.
  # Term ==36 and LoanMonthsSinceOrigination >= 40
  # In this case we see that return = Effective yield (no difference)
  # implying that estimated loss rate = 0.
  ggplot(aes(x = EstimatedEffectiveYield,y = EstimatedReturn),data = pd[pd$Term ==
  36 & pd$LoanMonthsSinceOrigination >= 39,]) +
  geom_point(color = 'red')
#See Fig.30
```

```
  # Term ==36 and LoanMonthsSinceOrigination < 39
  # There are cases where it is within 39 months of the loan's beginning data
  # i.e. no later than 3 months after loan term
  ggplot(aes(x = EstimatedEffectiveYield,y = EstimatedReturn),data = pd[pd$Term == 36 &
                                                        pd$LoanMonthsSinceOrigination < 39,]) +
  geom_point(color = 'red') + guides(color = guide_legend(title = "Rating", override.aes =
  list(alpha = 1,size = 5)))
#See Fig.31
```

```
  #
  ggplot(data = pd[pd$Term == 36 & pd$LoanMonthsSinceOrigination < 39,]) +
  geom_point(aes(x = EstimatedEffectiveYield,y = EstimatedReturn,color = ProsperScore)) +
  stat_smooth(aes(x = EstimatedEffectiveYield,y = EstimatedReturn),method = "lm",
              formula = y ~ poly(x,1), size = 1) +
  guides(color = guide_legend(title = "Score", override.aes = list(alpha =
```

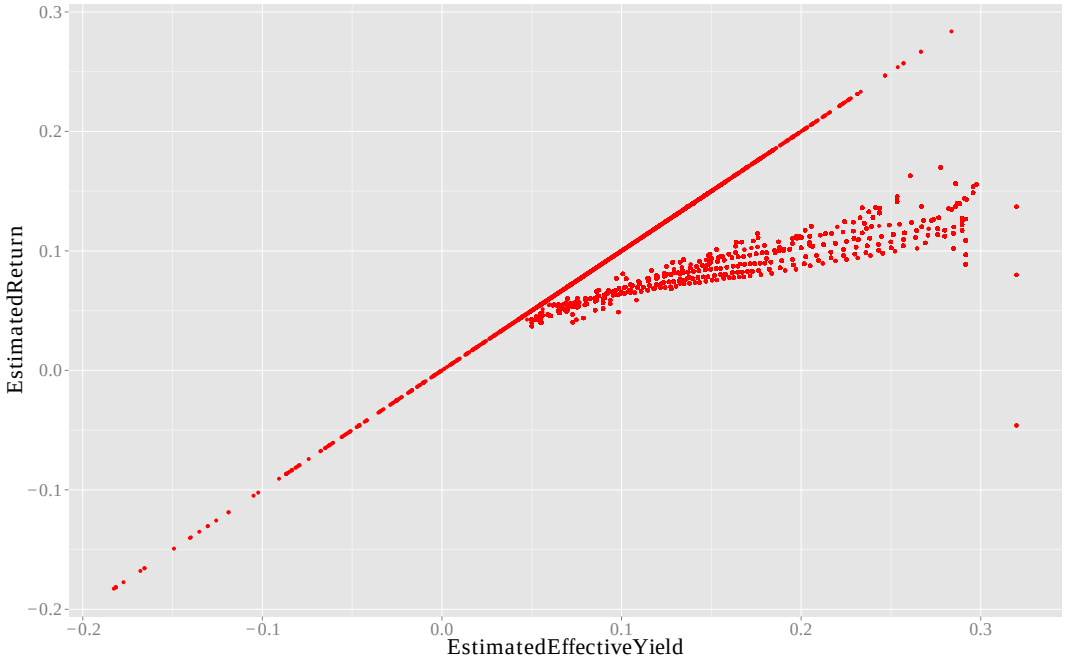


Figure 29: This is the same plot as above (Estimated Return vs. Effective Yield) shown only for loans where Term = 36 months. This plot has both different regioins of linear relationship between the variables.

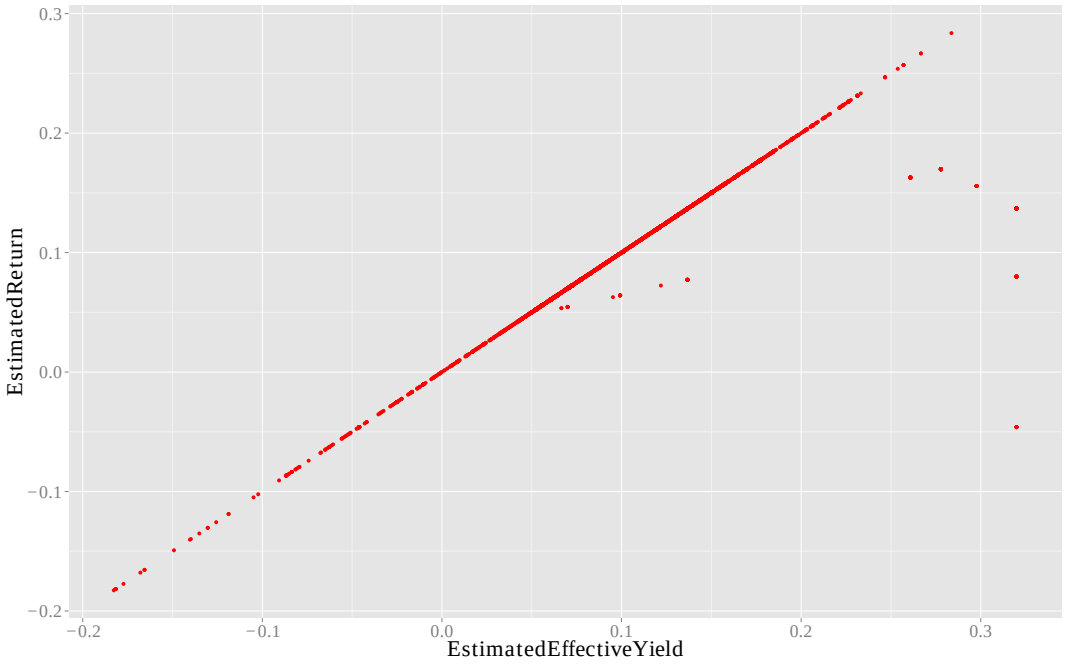


Figure 30: By plotting the Loans with Term = 36 months and LoanMonthsSinceOrigiation > 39, we are able to separate the two different zones in the data. We find that for such loans, the EstimatedReturn = EffectiveYield, implying that the loss is zero for this sub-category, since EstimatedReturn = EffectiveYield-EstiamtedLoss.

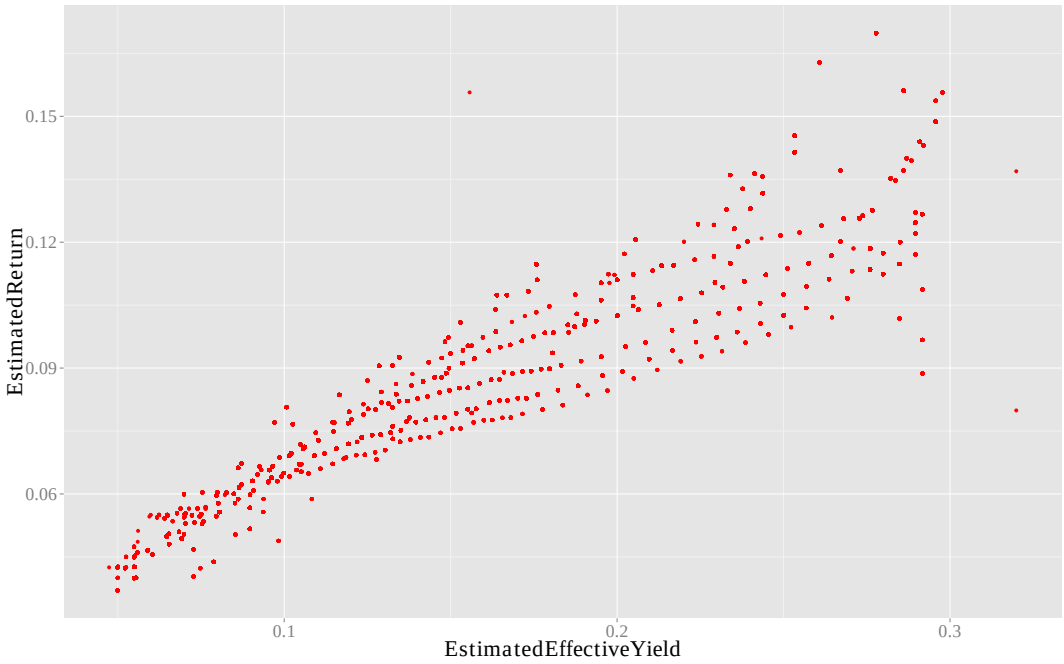


Figure 31: By plotting the Loans with Term = 36 months and LoanMonthsSinceOrigination < 39, we separate out the two different linearity regimes in the data.

```
  1,size = 5)))
#See Fig.32
```

```
  ######
  # We try another way to split the data, by classifying them as a loan that is on-going (more
  # payments are expected in the future) or a loan that is done (no more payments expected).
  #

  levels(pd$LoanStatus) # The different levels that a LoanStatus is in.
```

```
 [1] "Completed"              "Current"                "FinalPaymentInProgress"
 [4] "Past Due (1-15 days)"   "Past Due (16-30 days)"  "Past Due (31-60 days)"
 [7] "Past Due (61-90 days)"  "Past Due (91-120 days)" "Past Due (>120 days)"
[10] "Defaulted"              "Chargedoff"             "Cancelled"
```

```
  unique(pd$LoanStatus)
```

```
 [1] Completed              Current                Past Due (1-15 days)
 [4] Defaulted              Chargedoff             Past Due (16-30 days)
 [7] Cancelled              Past Due (61-90 days)  Past Due (31-60 days)
[10] Past Due (91-120 days) FinalPaymentInProgress Past Due (>120 days)
12 Levels: Completed < Current < ... < Cancelled
```

```
  pd$LoanStatusBucket = mapvalues(pd$LoanStatus, from = levels(pd$LoanStatus), to = c(
    "Done","OnGoing","OnGoing","OnGoing","OnGoing","OnGoing","OnGoing",
    "OnGoing","OnGoing","Done","Done","Done" ))
```

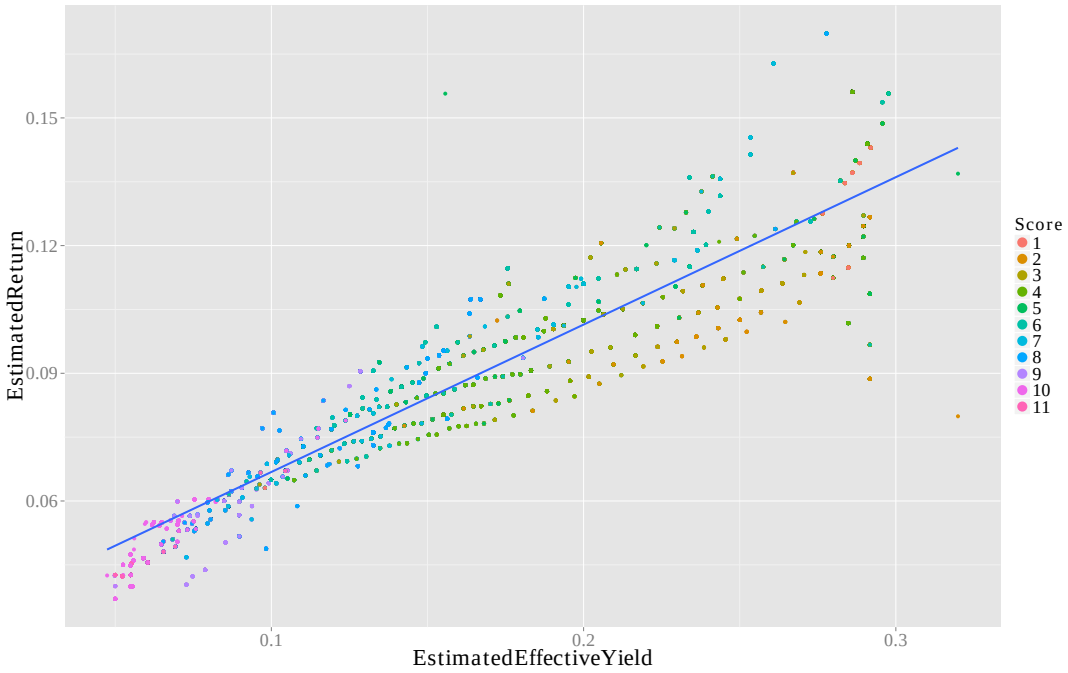


Figure 32: Coloring the Estimated Returns vs Yield plot by ProsperScore reveals that there is an increase in both Yield and returns based on worsening ProsperScores.

```
  #"Current","FinalPaymentInProgress","Completed","Past Due (1-15 days)","Past Due (16-30 days)",
 "Past Due (31-60 days)", "Past Due (61-90 days)", "Past Due (91-120 days)", "Past Due (>120 days)",
"Defaulted","Chargedoff","Cancelled"

  ggplot(data = pd[pd$Term == 36,]) +
  geom_point(aes(x = BorrowerRate,y = EstimatedLoss,color = LoanStatusBucket)) +
  guides(color = guide_legend(title = "Status", override.aes = list(alpha =1,size = 5)))
  # See Fig.33
```

```
  # Here it might seem that the linear relationship between the variables is when the status
  # is ongoing; else it seems like there is a wider spread of borrower rate for a given
  # EstimatedLoss
  ggplot(data = pd[pd$Term == 36 &
  pd$LoanStatusBucket == "Done",]) +
  geom_point(aes(x = BorrowerRate,y = EstimatedLoss,color = LoanStatusBucket)) +
  guides(color = guide_legend(title = "Status", override.aes = list(alpha = 1,size = 5)))
  # See Fig.34
```

```
  # We see that this kind of splitting does not clearly demarcate the linear and spreadout
  # regimes. This indicates that there must be some other criterion splitting the data in to the
  # two regimes.

  ###### By ProsperScore
  # Define a function to compute stats
  stat_sum_single <- function(fun, geom = "point", ...) {
  stat_summary(
```


Figure 33: We try another way of splitting the different linear regimes in the data by categorizing the loan data in to "On Going" or "Done".It seems that such a split also captures the two regimes cleanly. Here we look at EstimatedLoss vs BorrowerAPR.(both quantities affect yield and returns, the variables in the previous plots.)

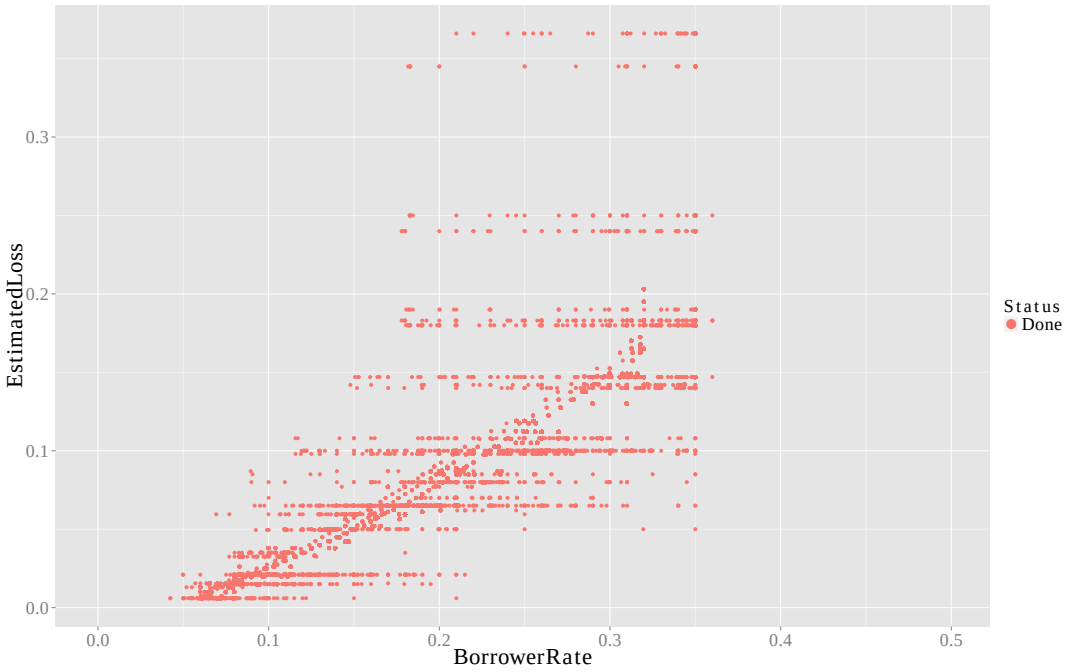


Figure 34: To be certain of this split along the Loan Status Bucket of "On-Going" or "Done" we plot only the loans that are Done with LoanTerm =36 months. We see that the two different regimes are still present and LoanStatusBucket is actually not a good way to split the two regimes in the data despite the apparent look from the previous plot.

```
fun.y = fun, colour = "black", geom = geom, size = 1, ...
)
}

p1 <-
ggplot(data = pd[pd$Term == 36 &
pd$LoanMonthsSinceOrigination < 39 & !is.na(pd$EstimatedEffectiveYield) ,],aes(x =
                                        EstimatedLoss,y = EstimatedEffectiveYield)) +
stat_sum_single(mean,geom = "line",linetype=2) +
stat_smooth(method = "lm", formula = y ~ poly(x,1), size = 1) +
geom_point(aes(color = ProsperScore),alpha = 1 / 2) +
guides(color = guide_legend(override.aes = list(alpha = 1,size = 5)))
print(p1)
```

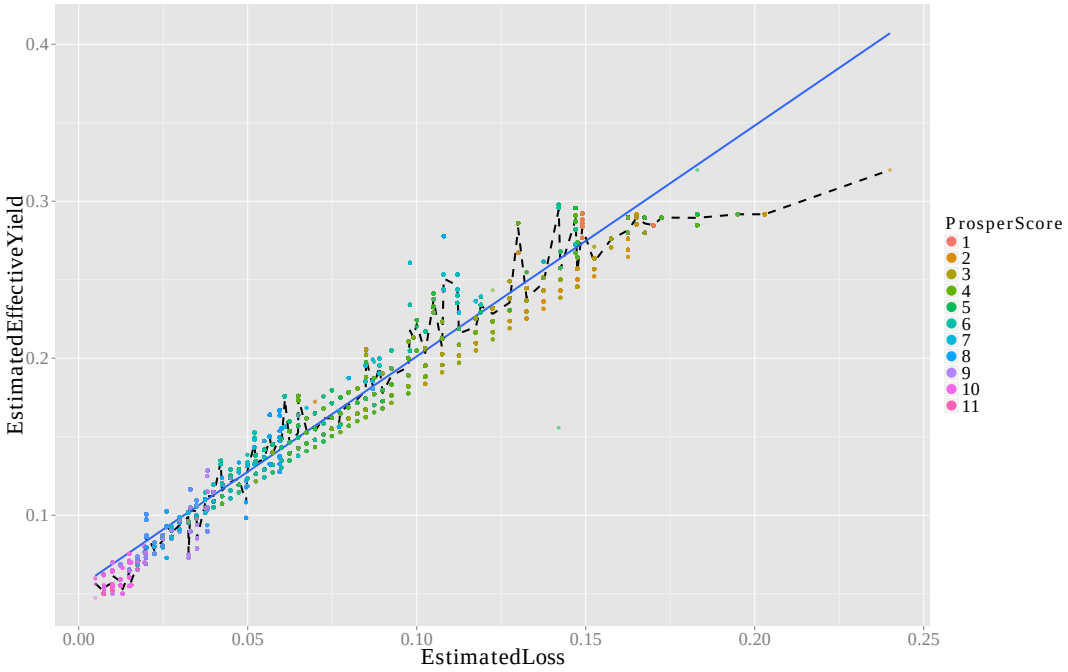


Figure 35: With the information from the above plots (splitting the data by Terms and LoanMonthsSinceOrigination) we plot other features like EffectiveYield Vs Loss and color it by ProsperScore. We see that those criteria (Term=36 months and LoanMonthsSinceOrigination >=39) do split the data in to separate regimes even while considering different features.

```
pd$Term = as.factor(pd$Term)
npd <- pd[!is.na(pd$EstimatedEffectiveYield) &
            pd$EstimatedEffectiveYield <0.275 & pd$LoanMonthsSinceOrigination >= 39
          & pd$Term == 36,]

p2 <-   ggplot(data = npd,
aes(x = EstimatedLoss,y = EstimatedEffectiveYield,color = ProsperScore,group=1)) +
geom_point(alpha = 1 / 2,size = 3)+
stat_smooth(method = "lm", formula = y ~ poly(x,2)) +
stat_sum_single(mean,geom = "line") +
guides(color = guide_legend(override.aes = list(alpha = 1,size = 5)))
print(p2)
```

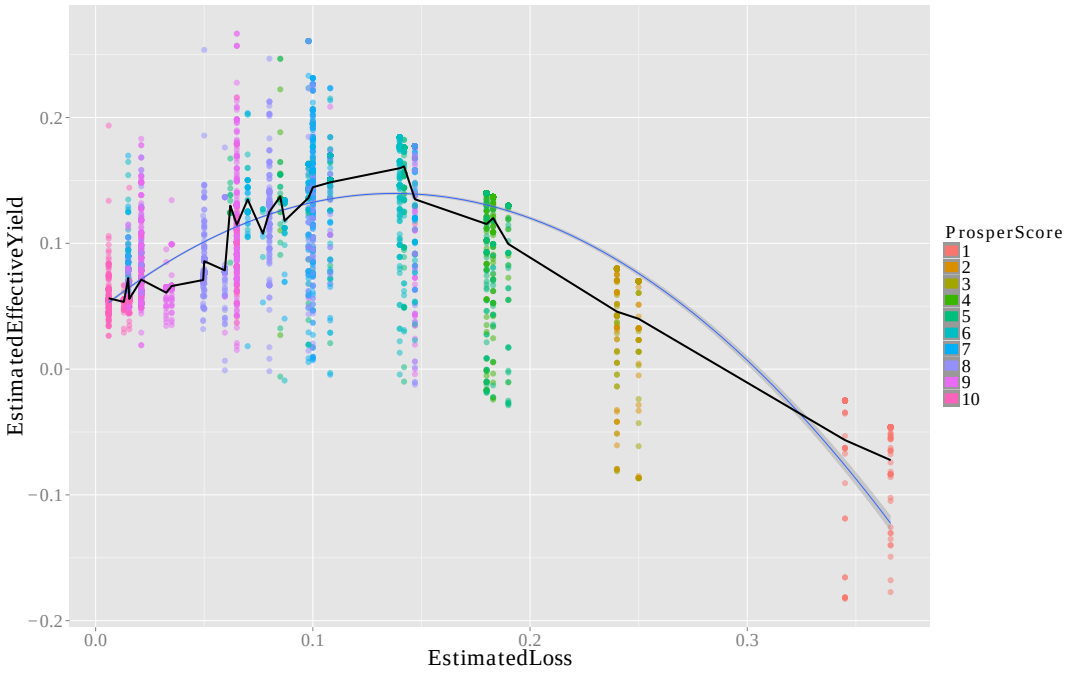


Figure 36: Here we show the Yield Vs. Los plot for loans with Term==36 months only. We see that trend in Loss is the same as other Terms, i.e. more loss with worsening ProsperScore, but the Yield increases until a ProsperScore of 7 and then drops as the Score decreases. We fit a quadratic polynomial to capture this trend, but looking at the mean EffectiveYield (black line) we can see that two straight lines would describe the trend much better.

```
  p3 <- ggplot(data = pd[!is.na(pd$EstimatedEffectiveYield) &
  pd$EstimatedEffectiveYield < 0.275 &
  pd$LoanMonthsSinceOrigination < 39,],aes(x = EstimatedLoss,y = EstimatedEffectiveYield,shape=
  Term))+
  stat_smooth(method = "lm", formula = y ~ poly(x,1), size = 1,aes(color = Term)) +
  guides(color = guide_legend(override.aes = list(alpha = 1,size = 8)))
  p4 <-   p3 + geom_point(aes(x = EstimatedLoss,y = EstimatedEffectiveYield,shape = Term),alpha =
  1 / 10, size = 3  ) + guides(shape = guide_legend(override.aes = list(alpha = 1,size = 4)))
  print(p4)
  # See Fig.37
```

```
#Reconvert EstimateLoss to a numeric type to allow math operations.
pd$EstimatedLoss = as.numeric(pd$EstimatedLoss)
pDtoIR<- ggplot(aes(y = DebtToIncomeRatio,x = BorrowerRate),data =
           pd[pd$Term == 36 & pd$DebtToIncomeRatio > 0 ,])+
        geom_point(aes(color = ConsolCreditGrade),position = position_jitter(width = 0.005,
                        height = 0.05),alpha=1/3) +scale_color_brewer(palette="Set1")+
        scale_y_log10(limits=c(0.005,10^0.4))+
        xlim(c(0,0.4))+
        stat_smooth(method = "lm", formula = y ~ poly(x,1),se=TRUE,size=1)+
        guides(color = guide_legend(title = "Grade", override.aes = list(alpha = 1,size = 5)))
print(pDtoIR)
  # See Fig.38
```

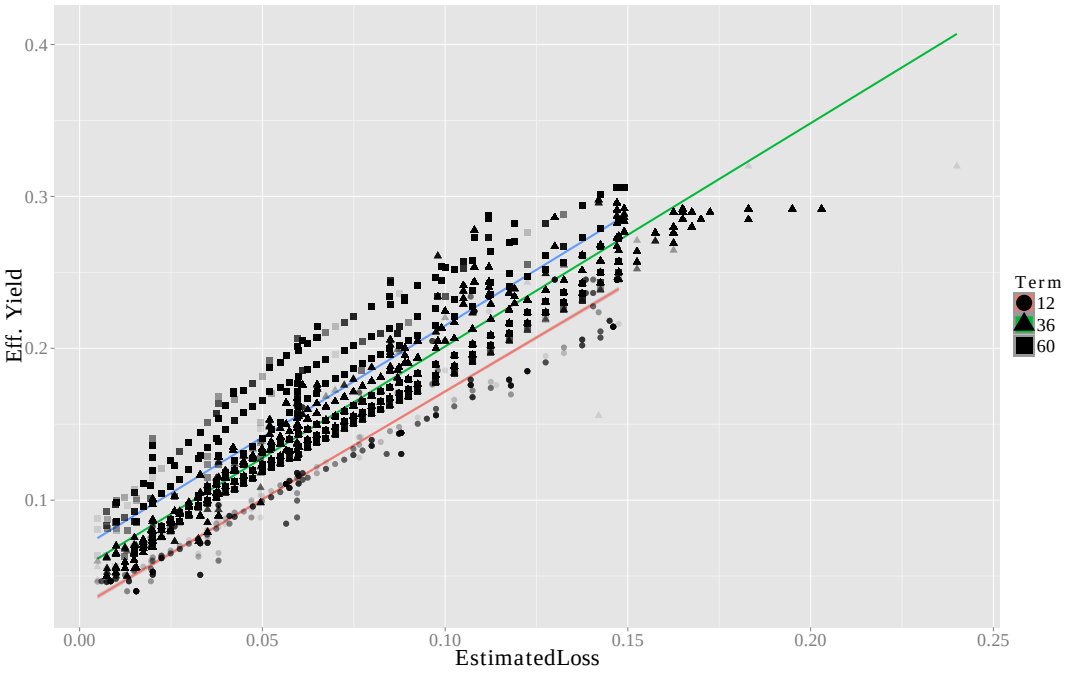


Figure 37: Here we show the linear regimes of all Terms (i.e. except Term ==36 and LoanMonthsSinceOrigiation >=39). Each Term is marked with a marker of a different shape and the trend is fitted with a straight line.


Figure 38: This plot shows the DebtToIncomeRatio Vs. Borrower APR. The data is colored by Credit Grade. We see that there is a gradual increase of the DebtToIncomeRatio as the BorrowerAPR goes up. The coloring of the data shows that Borrowers with higher DebtToIncomeRatio get worse CreditGrades and are hit with higher BorrowerAPR

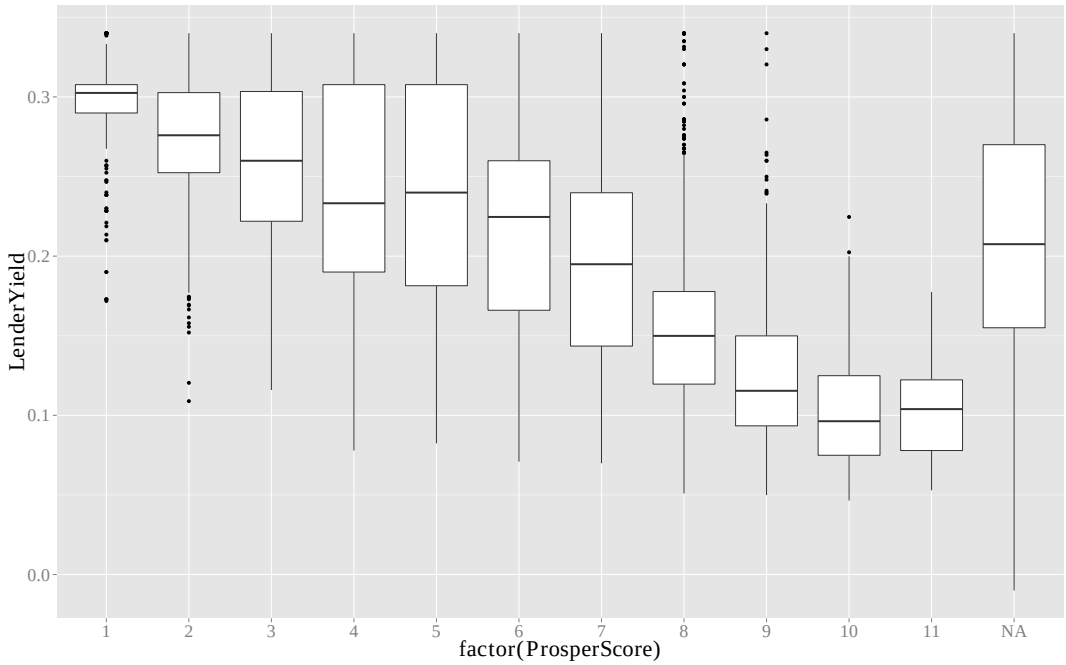


Figure 39: Plot of Lender Yield Vs. ProsperScore. We see that the mean Lender Yield goes down as the ProsperScore improves.

```
ggplot(aes(factor(ProsperScore),LenderYield),data=pds)+geom_boxplot()
 # See Fig.39
```

## Credit Score

```
# Histogram of credit scores - to guage the range
ggplot(aes(x=CreditScoreRangeLower),data=pd,binwidth=5)+geom_histogram()
# #See Fig.40
```

In what follows, we shall try to look at the EsitmatedLoss Vs. BorrowerRate, along the same lines as the considerations above; except, here will color them by CreditScoreRangeLower.

```
# EstimatedLoss plots, colored by CreditScore
ggplot(aes(y=EstimatedLoss,x=BorrowerRate,color=CreditScoreRangeLower),
       data=pd[pd$CreditScoreRangeLower>320,])+geom_point(alpha=1/2,size=3)+
scale_color_gradientn(colours = rainbow(10))+
guides(color=guide_colorbar(title="Cred Score", override.aes =list(alpha=1)))
# #See Fig.41
```

```
# Only the Term=36 month loans
ggplot(aes(y=EstimatedLoss,x=BorrowerRate,color=CreditScoreRangeLower),
       data=pd[pd$Term==36 & pd$CreditScoreRangeLower>320 & pd$LoanMonthsSinceOrigination >39,])+
geom_point(alpha=1/3,size=3)+
scale_color_gradientn(colours = rainbow(10))+
```

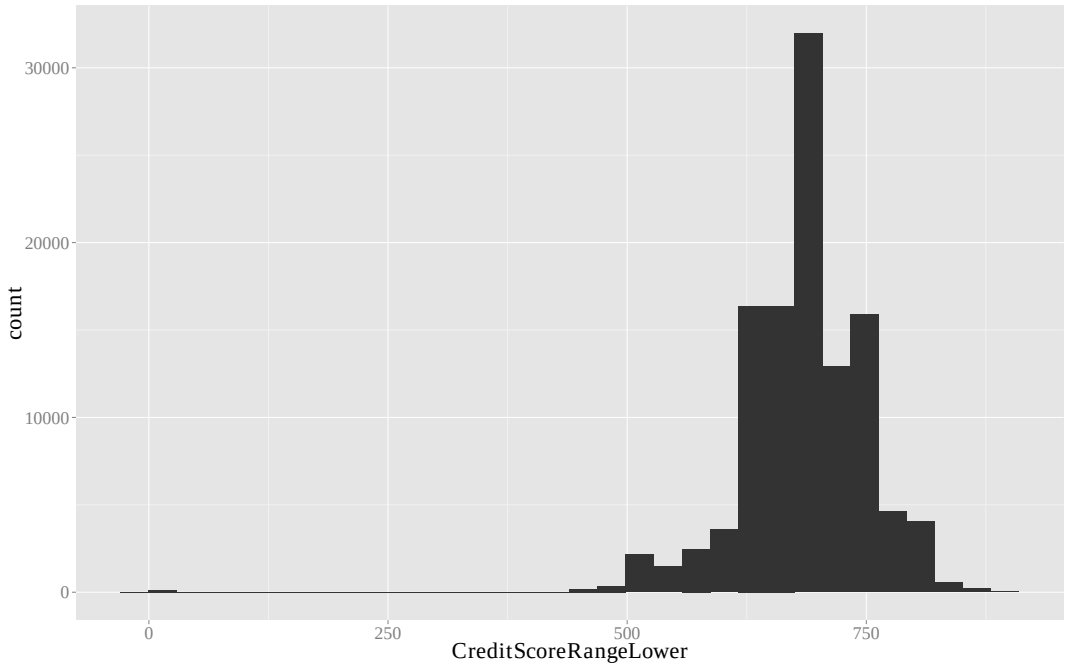


Figure 40: A histogram of the CreditScoreRangeLower shows that most scores are above 350 and range up to 800. There are a few Borrowers with 0 as their score also. This must be dure to lack of data that some were erroneously assigned such a low value.

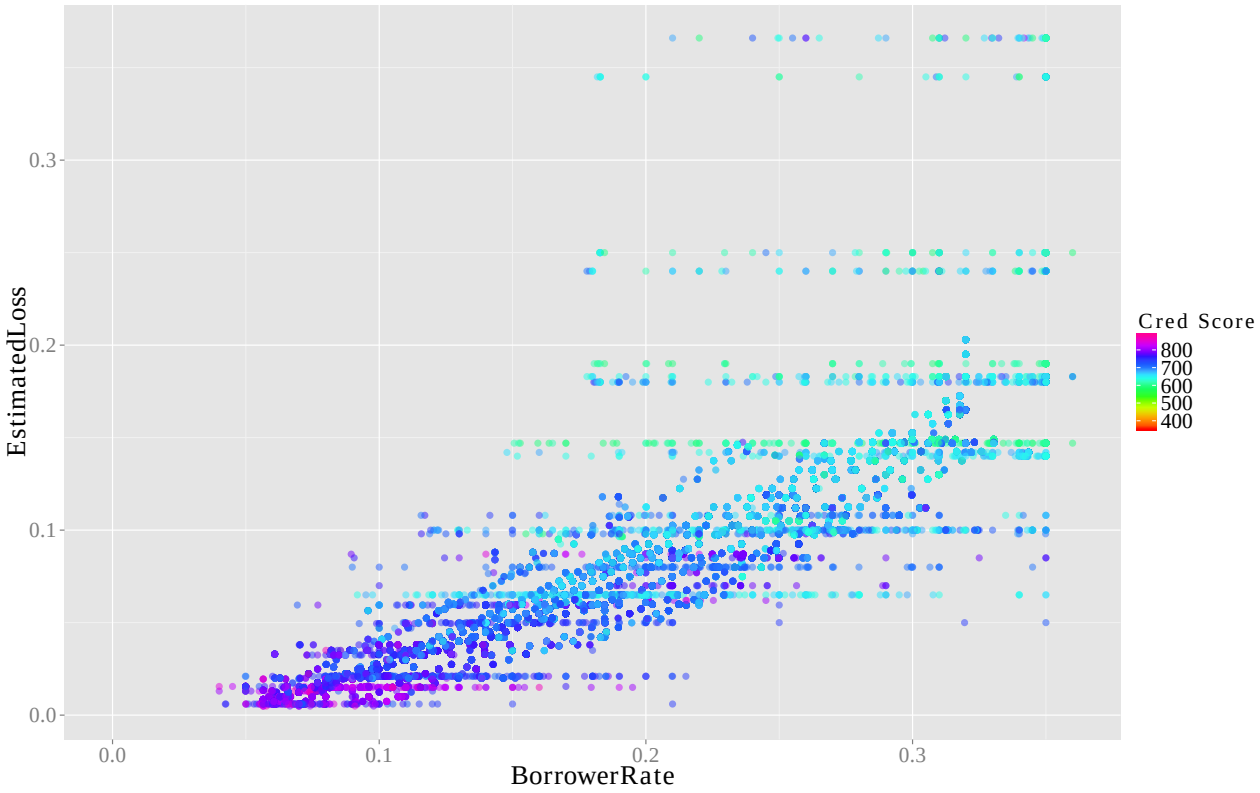


Figure 41: A plot of the EstimatedLoss Vs The BorrowerAPR, colored by the credit score. We can see that there is an increase in the Loss and APR as the credit score gets lower; however, the plot is too crowded.

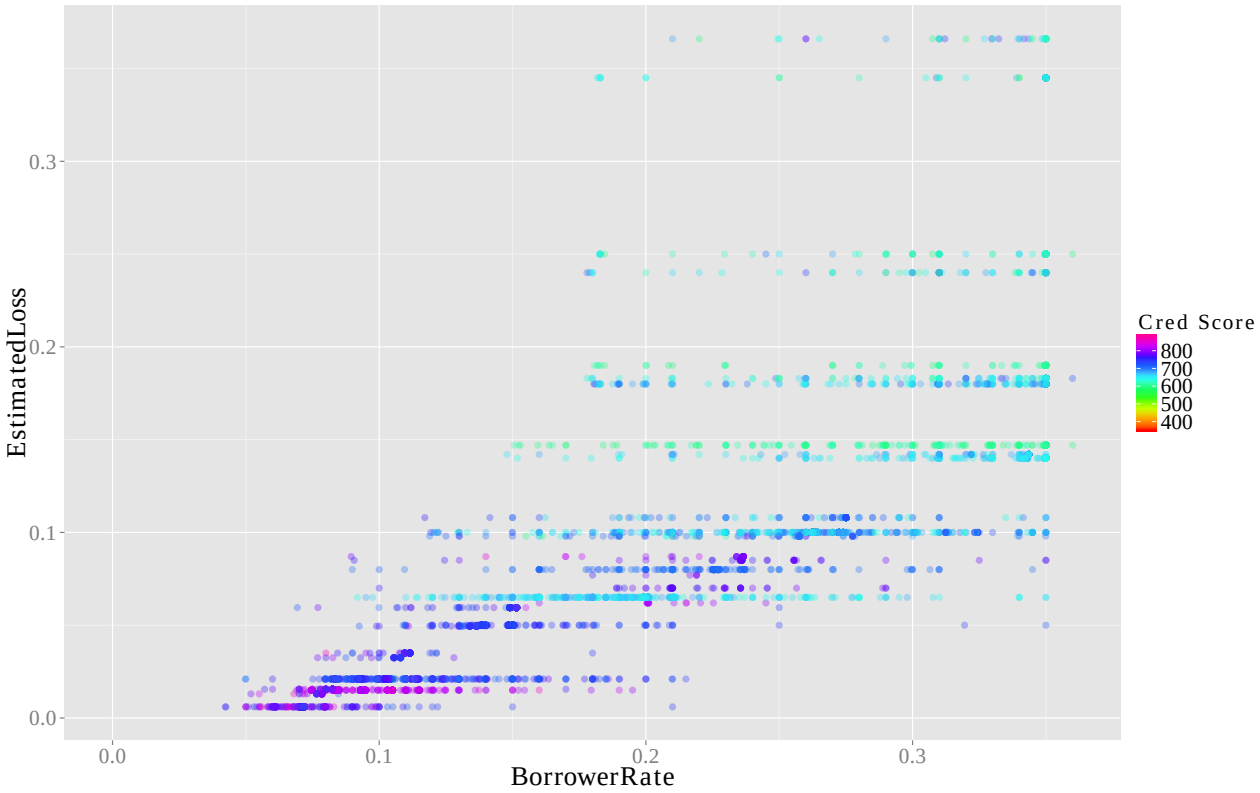


Figure 42: Here we show only loans with Term == 36 months and the LoanMonthsSinceOrigiation >39. Clearly the Loss rate increasee with lower CreditScore. The mean Borrower rate also increases with lower CreditScore.

```
guides(color=guide_colorbar(title="Cred Score", override.aes =list(alpha=1)))
# #See Fig.42
```

```
# Only the 12 and 60 month loans
library('gridExtra')
p1<-ggplot(aes(y=EstimatedLoss,x=BorrowerRate),data=pd[pd$CreditScoreRangeLower>320 &
                                                       pd$Term==12,])+
  geom_point(aes(color=CreditScoreRangeLower),alpha=1/3,size=3)+
geom_smooth(method="lm")+scale_color_gradientn(colours = rainbow(10))+
ggtitle("Term=12 Months")+
guides(color=guide_colorbar(title="Cred Score", override.aes =list(alpha=1)))
# #See Fig.43
p2 <- ggplot(aes(y=EstimatedLoss,x=BorrowerRate),data=pd[pd$Term==60,])+
geom_point(aes(color=CreditScoreRangeLower),alpha=1/3,size=3)+
scale_color_gradientn(colours= rainbow(10))+
geom_smooth(method="lm")+ggtitle("Term =60 Months")+
guides(color=guide_colorbar(title="Cred Score", override.aes =list(alpha=1)))
# #See Fig.43
p3 <- ggplot(aes(y=EstimatedLoss,x=BorrowerRate),data=pd[pd$Term==36
                                                        &pd$LoanMonthsSinceOrigination <39 ,])+
geom_point(aes(color=CreditScoreRangeLower),alpha=1/3,size=3)+
scale_color_gradientn(colours= rainbow(10))+
geom_smooth(method="lm")+ggtitle("Term =36 Months (Loan Month <39)")+
```

```
guides(color=guide_colorbar(title="Cred Score", override.aes =list(alpha=1)))

grid.arrange(p1,p3,p2,ncol=1)
# #See Fig.43
```

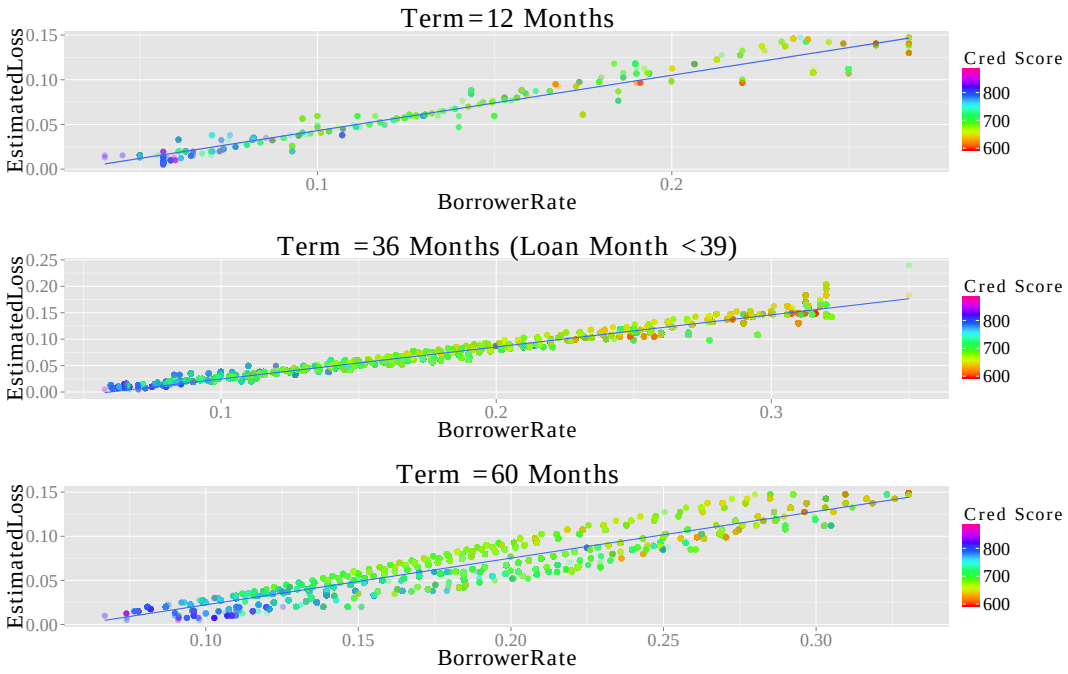


Figure 43: Here we show the EstimatedLoss Vs. Borrower rate for all the three loan Terms. For loans with Term = 36 months, we only plot those loans with LoanMonthsSinceOrigination < 39 months. They all show a linear trend and both variables increase with lowering CreditScores.

```
# All three time TERM periods (12, 36 and 60 months)
ggplot(aes(y=EstimatedLoss,x=BorrowerRate,color=Term),data=pd)+geom_point(alpha=1/5,size=3)+
scale_color_brewer(palette = 'Set1',guide = guide_legend(title = 'Term',
                   reverse = T,override.aes = list(alpha = 1, size = 5)))
# #See Fig.44
```

```
#Plot by loan-status and color by credit score - bar plot, log-scale
pd$EstimatedLoss = as.numeric(pd$EstimatedLoss)
ggplot(aes(y=EstimatedLoss,x=LoanStatus,color=CreditScoreRangeLower),
       data=pd[pd$CreditScoreRangeLower>400,])+
geom_point(alpha=1/5,size=3)+scale_y_log10()+
geom_boxplot(outlier.size = .1,alpha=1/20)+
theme(axis.text.x  = element_text(angle=45, vjust=1.0, size=15))+
guides(color=guide_colorbar(title="Cred Score", override.aes =list(alpha=1)))+
scale_color_gradientn(colours=rainbow(10))
# #See Fig.45
```

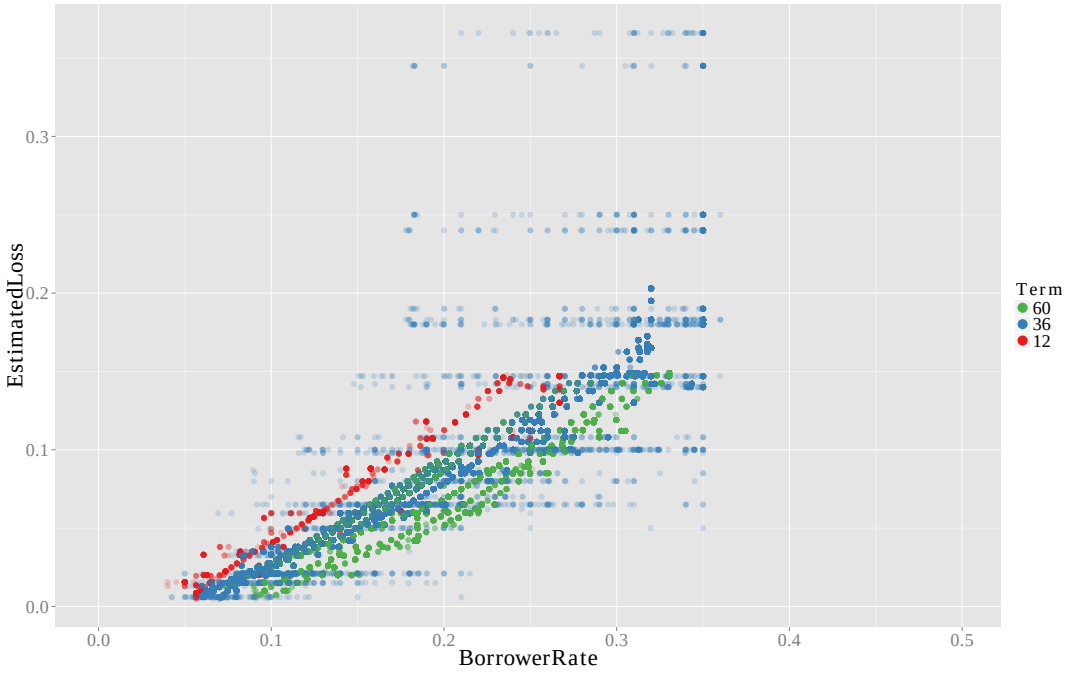


Figure 44: Showing the Estimated Loss Vs. Borrower Rate plot with different colors showing the different Terms. We can clearly see that only Term==36 months have the two distinct regimes in the data.

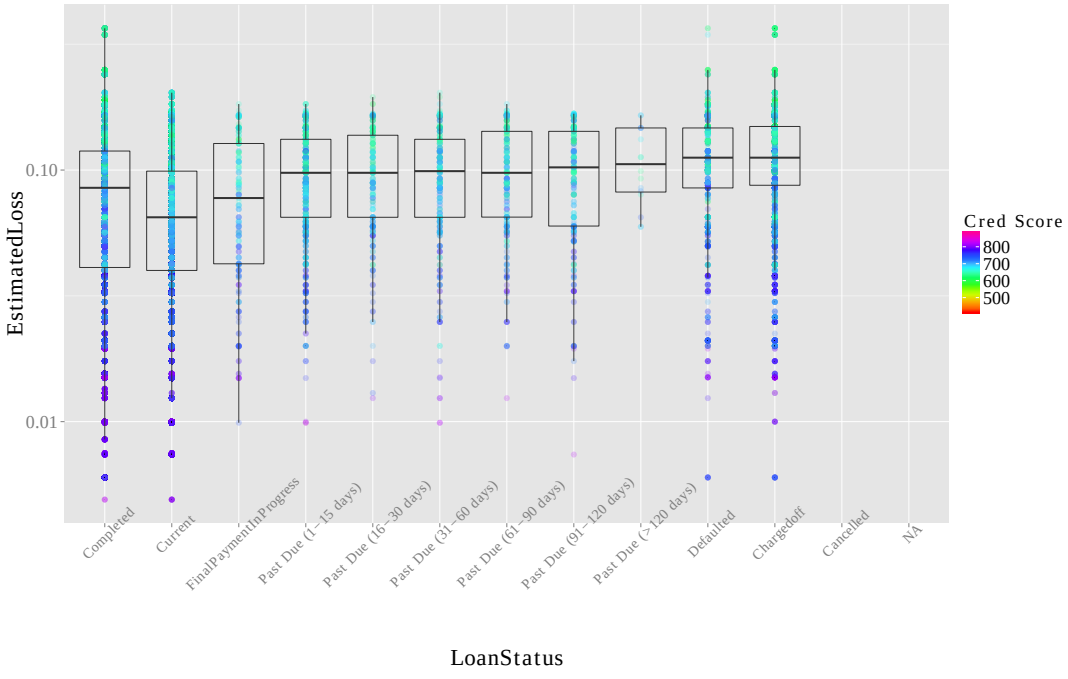


Figure 45: A bar chart showing the EstimatedLoss for different LoanStatus and coloring the points by Credit Score. We see that for loans other than the ones that are completed, there is a steady increase in the mean EstimatedLoss. For every Loan Status the loans made to Borrowers with lower Credit Score make more loss.

## Credit Lines

```
# The Effect of # of credit lines on the EstimatedLoss and EstimatedReturn
pd$CreditLines.bucket <- cut(pd$CurrentCreditLines, breaks =c(0,seq(1,25,5),60))

ggplot(aes(y=EstimatedLoss,x=LoanStatus,color=CreditLines.bucket),data=
                                  pd[!is.na(pd$EstimatedLoss) & !is.na(pd$CreditLines.bucket),])+
      scale_color_brewer(type='div',palette = 'Set1',guide =
                                              guide_legend(override.aes = list(alpha=1,size=1)))+
      geom_boxplot(outlier.size = .5)+ylim(0,.25)+
      theme(axis.text.x  = element_text(angle=45, vjust=0.8, size=18))+
      guides(color=guide_legend(title="Cred Lines", override.aes =list(alpha=1)))
#See Fig. 46
```

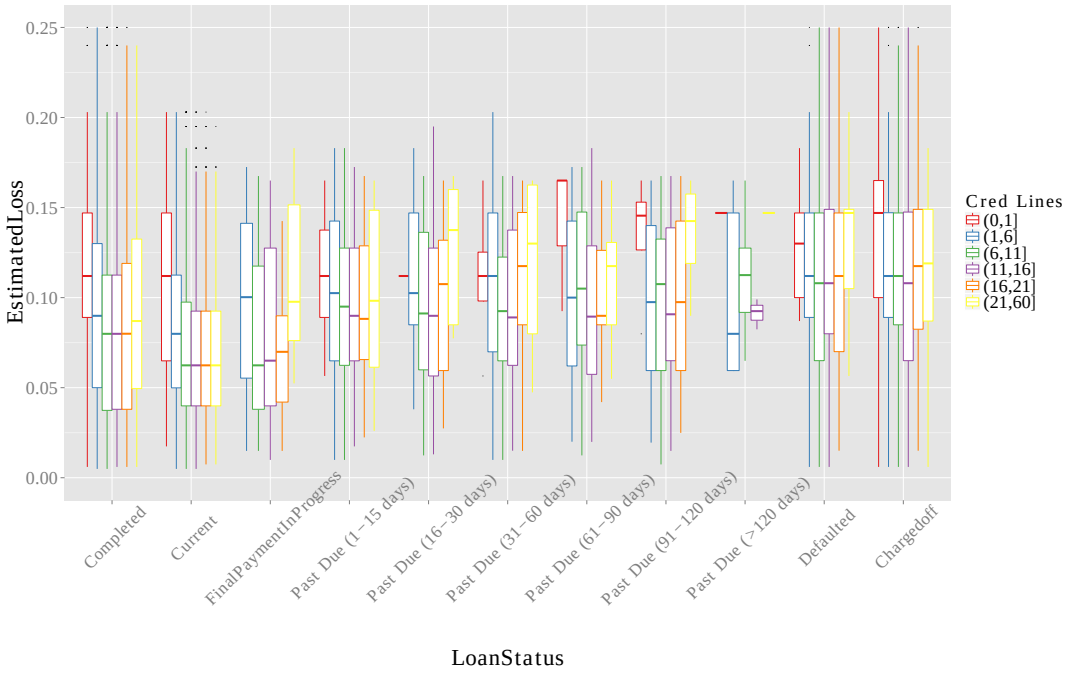


Figure 46: We try to see the effect the number of lines of credit that a borrower has on the Estimated Loss. We break the number of credit lines in to buckets and notice that for every loan status the Loss is highest for loans with least credit lines. It decreases with increasing lines of credit and then further starts to increase above 20 lines of credit.

```
p0<-ggplot(aes(y=EstimatedReturn,x=LoanStatus,color=CreditLines.bucket),data=
                                  pd[!is.na(pd$EstimatedLoss) & !is.na(pd$CreditLines.bucket),])
p1 <- p0+ scale_color_brewer(type='div',palette = 'Set1',guide =
                               guide_legend(override.aes = list(alpha=1,size=1)))+
      geom_boxplot(outlier.size = .5)+ylim(0,.25)+
      theme(axis.text.x  = element_text(angle=45, vjust=0.8, size=18))+
      guides(color=guide_legend(title="Cred Lines", override.aes =list(alpha=1)))
print(p1)
#See Fig. 47
```

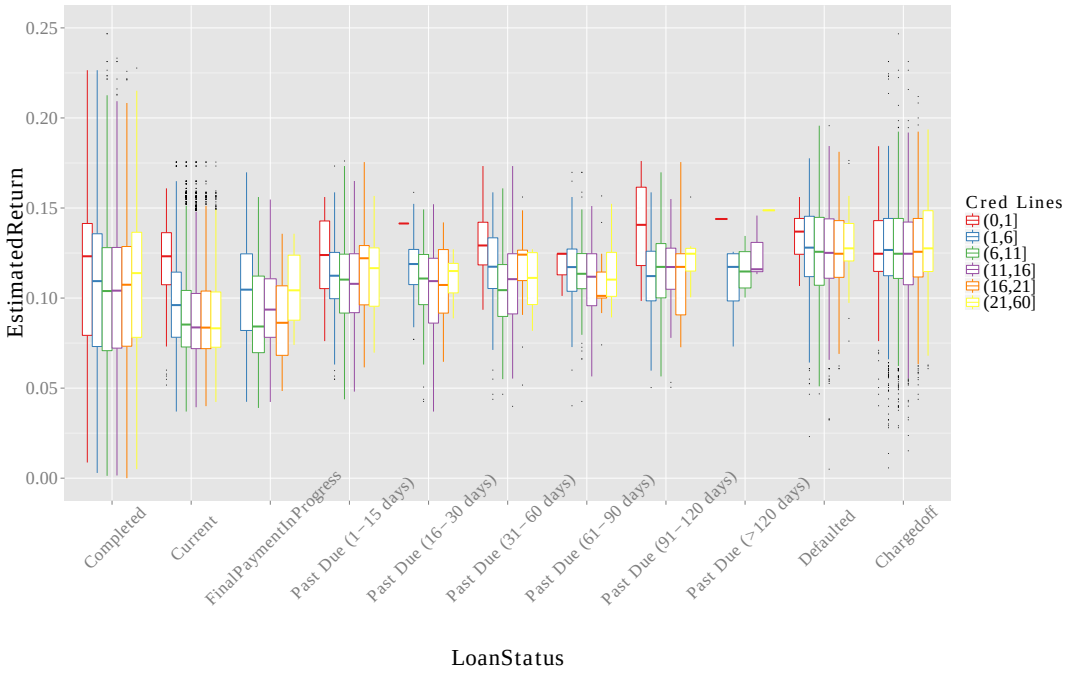


Figure 47: A similar plot as above but seeing the dependance of Estimated Returns on number of lines of credit. The returns are high for lower number of credit lines, it then decreases with increasing credit lines and again increases beyond 20 lines of credit. This happens for all the loan statuses.

```
pd$ReturnCut <- cut(pd$EstimatedReturn, breaks =c(-0.2,-1e-9,seq(0,0.3,0.01)))
ggplot(aes(x=CurrentCreditLines),data=pd[!is.na(pd$ReturnCut) & !is.na(pd$CurrentCreditLines),])+
 geom_histogram(binwidth=1,aes(fill=ReturnCut,y=..count..))+scale_x_discrete()+
theme(axis.text.x=element_text(size=15))+guides(fill=guide_legend(title="Returns"))
#See Fig. 48
```

## Customer Payments

```
Customer Payments Vs. Original Loan Amount (#See Fig.49)
 ggplot(aes(y=LP_CustomerPayments,x=LoanOriginalAmount),data=
          pd[pd$LP_CustomerPayments>1,],binwidth=.01)+
  geom_point(alpha=1/10)+ylab('Customer Payments')+
  xlab('Loan Original Amount')
CommentTok# See Fig. 49
```

In the initial scatterplot matrix (Fig.3) we see that in the LoanMonthsSinceOrigination column (column 4 ) there is a gap around month=60. To investigate this further we make a few plots.

```
Payment Histograms show gap in LoanMonthsSinceOrigination (#See Fig.50)
set.seed(1000)
pdn <- pd[sample(1:length(pd$LenderYield),7000),]
p1 <- ggplot(aes(x=LoanMonthsSinceOrigination,y=LP_CustomerPayments),data=pdn)+
  ylab("Cust. Paymt")+
  geom_point(aes(color=Term))+ggtitle('All Terms')
```

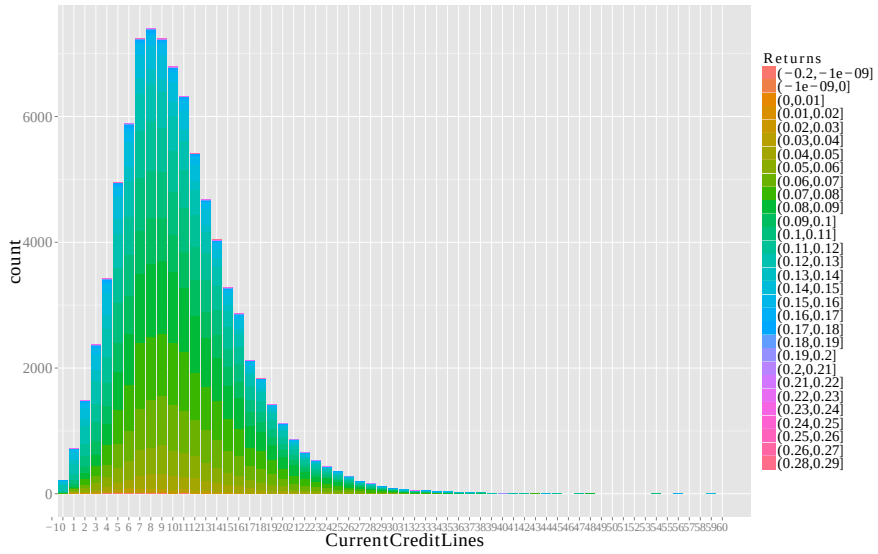


Figure 48: To test if we can see the trend shown in the previous plot (Fig.47) when we look at returns without breaking the credit lines in to buckets or by slicing the data by loan status, we plot a histogram of the returns and color the histogram by the returns. We are unable to find a particular trend in the returns. However we notice that for all credit lines the return rates between 0.05 and 0.15 form bulk of the loans. The trend seen in the previous plot is lost when all the loan status are considered together. (It would be great if one could color the histogram without having to break the returns in to buckets.)

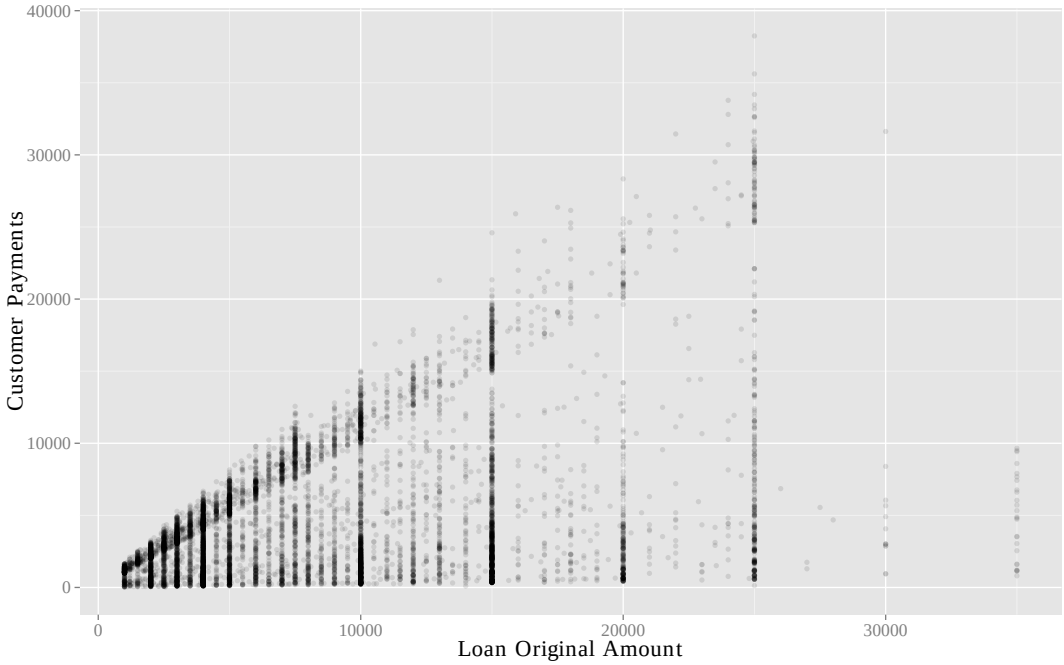


Figure 49: We can see that some of the loans show Customer Payments above the loan amount. These are probably the loans that have been completed and the amount above the Loan Amount indicates the Returns for the lender. Many other loans are below the line with slope =1 , indicating the customer payments have not yet broken even with the Original Loan amounts. These are most likely loans that are in progress (i.e. have not been completed,defaulted,charged off, or is in FinalPaymentInProgress.)

```
p2 <- ggplot(aes(x=LoanMonthsSinceOrigination,y=LP_CustomerPayments),data=pdn[pdn$Term==36,])+ylab("Cus
  geom_point(aes(color=Term))

p3 <- ggplot(aes(x=LoanMonthsSinceOrigination,y=LP_CustomerPayments),data=pdn[pdn$Term==36,])+ylab("Cus
  geom_point(aes(color=EstimatedLoss))+ggtitle('By Estimated Loss ')

p4 <- ggplot(aes(x=LoanMonthsSinceOrigination,y=LP_CustomerPayments),data=pdn[pdn$Term==36 & !is.na(pdn
  geom_point(aes(color=EstimatedLoss))+ggtitle('Non NA values')

p5 <- ggplot(aes(x=LoanMonthsSinceOrigination,y=LP_CustomerPayments),data=pdn[pdn$Term==36 & is.na(pdn$
  geom_point(aes(color=Term))+ggtitle('Term = 36 (NA values)')

p6<-ggplot(aes(x=LoanMonthsSinceOrigination,y=LP_CustomerPayments),data=pd[pd$Term==36 & is.na(pd$Estima
  guides(color = guide_legend(title = "Loan Status", override.aes = list(alpha = 1,size = 5)))

grid.arrange(p1,p2,p3,p5,p4,p6,ncol=2)
```

```
length(pd[is.na(pd$EstimatedLoss),]$EstimatedLoss) # No.of NA values.
```

```
[1] 29084
```

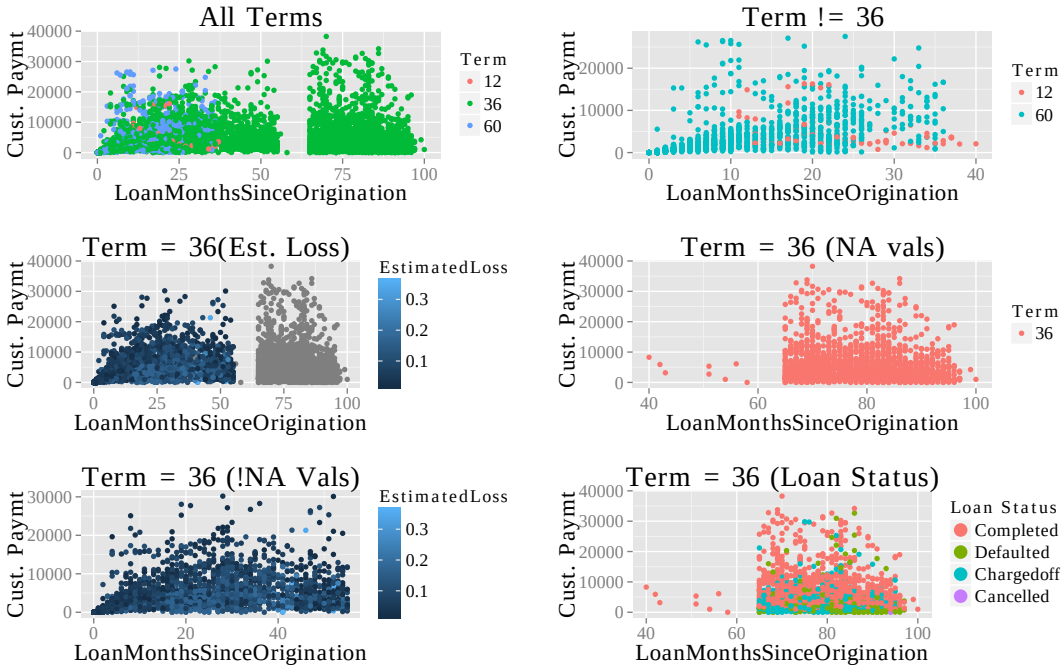


Figure 50: Progression of Analysis showing how we narrow down that the gap in the LoanMonthsSinceOrigination column in the 3rd scatter plot (Fig.3), near month=60. That is, the loans that have crossed 60 months since origination have been completed,defaulted,charged-off, or cancelled. So their EsitmatedLoss/Returns/Yield fields hold NA values.

### Equations

I introduce a few notations to make it slightly easier to understand the relationship between the different variables based on their description:

$y_e$ = Effective Yield
$y_l$ = Lender Yield
$r$ = Returns
$I$ = Borrower Interest Rate
$f_s$ = Service Fees
$f_l$ = Late Fees
$u$ = uncollected interest on charge-offs
$l$ = Loss

$$y_e = I + f_l - f_s - u \tag{1}$$
$$r = y_e - l \tag{2}$$
$$y_l = I - f_s \tag{3}$$
$$\Rightarrow y_e = y_l + f_l - u \tag{4}$$
$$r = y_l + f_l - l - u \tag{5}$$

# Final Plots and Summary

## Returns By ProsperScore (and By Term)

In this plot (Figure:51) we look at the Estiamted Return for loans and analyze them based on Terms and colored by the ProsperScore. We find that in all plots there is a trend of decreasing returns with decreasing (i.e. better) ProsperScore. As a borrower this means that if one is looking to get a loan with the lowest interest rate, it is essential to have established really good credit worthiness. We see that, in reality, very few borrowes are able to maintain a pristine score. The vast majority of the lenders have a score of 8 or below. We would imagine that increasing risk of lending to borrowers with worse scores would prevent lending. However we see that even borrowers with a Score of 1 are able to secure loans. Not only that, the lender is also able to extract a higher return from these groups probably in the form of late fee and interest. Conversely if the return is low on better scores what could be the incentive to give out loans to such a group? As we will see in the following plot, such a group also minimizes loss and hence is less risky from the lender's perspective.

The 36 month long loans for which borrowers have taken more than 90 days since the loan period expires (i.e. LoanMonthsSinceOrigination >39), some loans show negative returns. For all other categories the lender always has positive returns on the loans made. *The Lender almost always makes a profit.* We also note that in this category, the number of borrowers with better scores increases dramatically over the other sections. Also the highest ProsperScore of 11 is not given to any borrower who has not repaid the loan within the term period (see legend in 4th subplot, 11 is missing). We see that the proportion of borrowers with a ProsperScore of 9 or higher is greatest in this category (Term =36 and LoanMonthsSinceOrigination > 39) compared to other categories.

## Returns By LoanStatus / Yield vs Loss

In this figure (Figure:52)we try to see if the number of credit lines a borrower has, has any effect on the returns the lender gets. From the histogram it is hard to tell because each credit line seems to have borrowers yielding returns at all levels (all colors). By considering the mean returns we see that there is in fact a reduction of returns based on the number of credit lines. Mean returns decrease from 0 to 7 credit lines and there after remains more or less constant till about 30 lines and then shows fluctuations. However by slicing the data based on the emphstatus of the loan we can see that the mean Returns does indeed decrease based on the number of credit lines for each Loan Status. First it decreases and then beyond 20 lines of credit, the

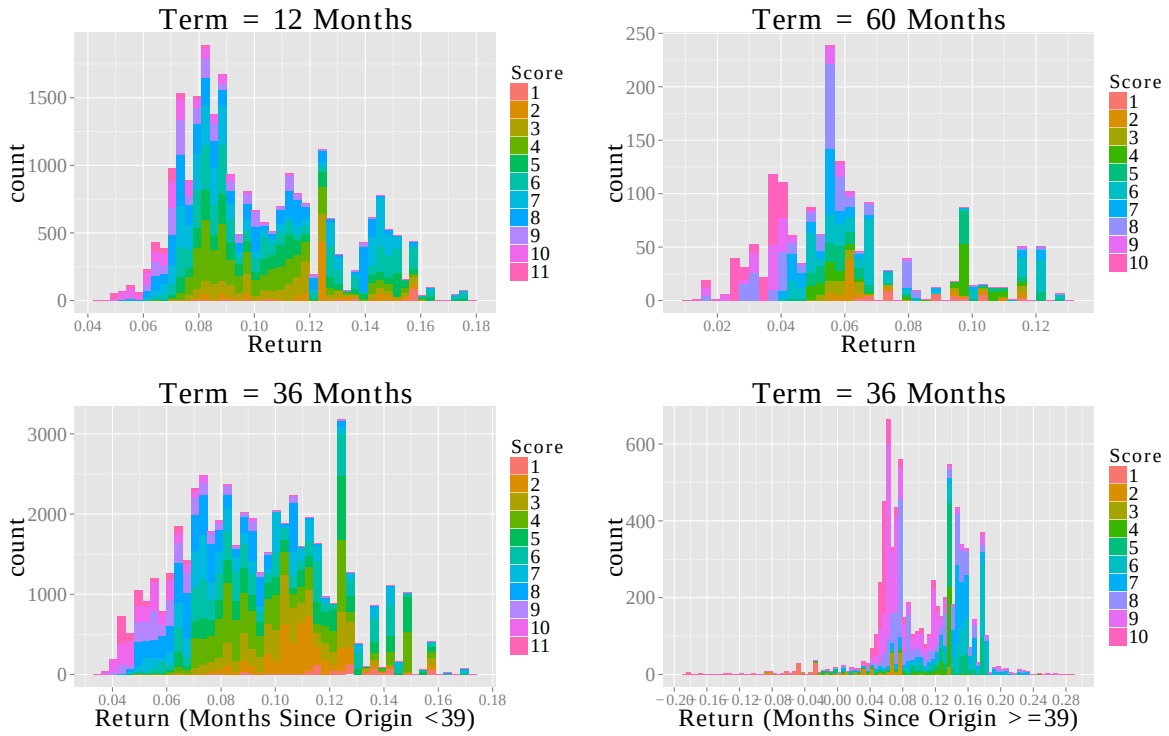


Figure 51: This figure shows the histogram of EstimatedReturns for various Terms (12,36,60 months). For the 36 month Loans we plot two different plots, one with LoanMonthsSinceOrigination <39 and LoanMonthsSinceOrigination >=39 months.

mean Returns again rises. Discounting Chargedoff loans all other categories show that there is considerable decrease in the mean returns beyond the 5 credit line bucket before rising again after 20 lines of credit.

Since the Returns is linearly related to Loss and Yield (see above equation), we plot the Yield vs Loss to assess if the lower returns (for better Prosper Scores) are as a result of greater loss or lower yields or both. We notice that, surprisingly, the Yield and Loss are related in an almost linear fashion and this holds for all ProsperScore groups. This linear trend is violated only for loans with 36 month Term that have passed the 90 limit since ending the Term limit (i.e. 39 months after Term origination).

## Trends In The Variables

Here (Figure:53)we explore some of the linear trends in the data that might not be obvious at first sight. EstimatedYield and EstimatedLoss both show a linear decrease with improving ProsperScore. We see that all the three Loan Terms show that this linear trend holds. However, even within a Term period, for example the 60 month loans, we see that for the same Loss there are multiple values of returns. This could be because of the difference in Interest rate, late fess, Service Fees or uncollected interest on charge-offs. To establish this we see that when colored by the APR the returns increase as the APR increases for a given EstimatedLoss. Thus the trend of increasing return with Borrower APR holds good for each value of Estimated Loss. Also we see that there are trends in the DebtToIncomeRatio (based on creditRating). The average DebtToIncomeRatio increases with worsening creditGrade. This is associated with an increase in the BorroweRate. The AmountDelinquent (log) increases with number of delinquencies (log) the borrower has had in the past 7 years. This relationship cannot be discerned in regular coordinates (i.e. without the log-scale).

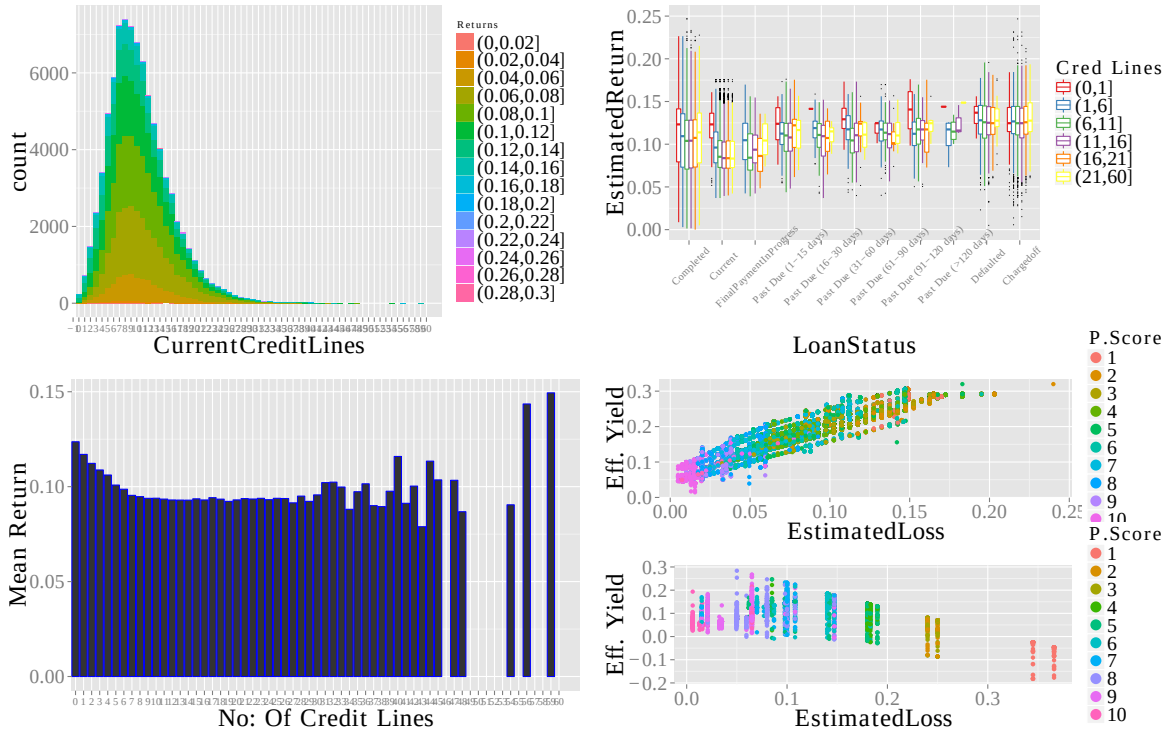


Figure 52: Here we show Estimated Returns and how that variable is affected by the number of credit lines, Loan Status, and Prosper Score. The ProsperScore effect is shown on Yield and Loss and since Returns = Yield-Loss, we can infer ProsperScore's effect on returns also.

## Summary

- The discovered trends seem to reflect intuitive thinking that as the credit worthiness of the borrower improves (ProsperScore, CreditScore, CreditRating) the BorrowerRate also goes down.
- Both Yield and Loss (for the lender) also decrease linearly with improving credit rating.
- The lender gets greater returns by lending to borrowers with mediocre and poor credit ratings (compared to people with excellent ratings) and recouping more money than is actually lent in the form of interest and fees.
- For a given loan status the number of creditlines a borrower seem to affect the returns yielded: the lesser the credit lines, higher the returns. When the returns are not split along loan status, this reduction of returns with increasing credit line is observed only up 7 lines of credit and then the return are more or less constant.
- The mean amount delinquent (log) on a loan is related to the number of delinquencies (log) in the last 7 years. The relationship is linear on the log-log scale.
- Th mean DebtToIncomeRatio (log) shows a linear relationship with to the BorrowerRate. In other words, the BorrowerRate levied on a borrower is affected by the DebtToIncomeRatio of the borrower.
- The EstimatedLoss increases with worsening credit rating.
- The EstimatedLoss for each LoanStatus increases as the credit score/rating lowers.
- Yield and Loss are linearly related and implies that returns and loss will also be linearly related (from the linearity of the equation listed above).

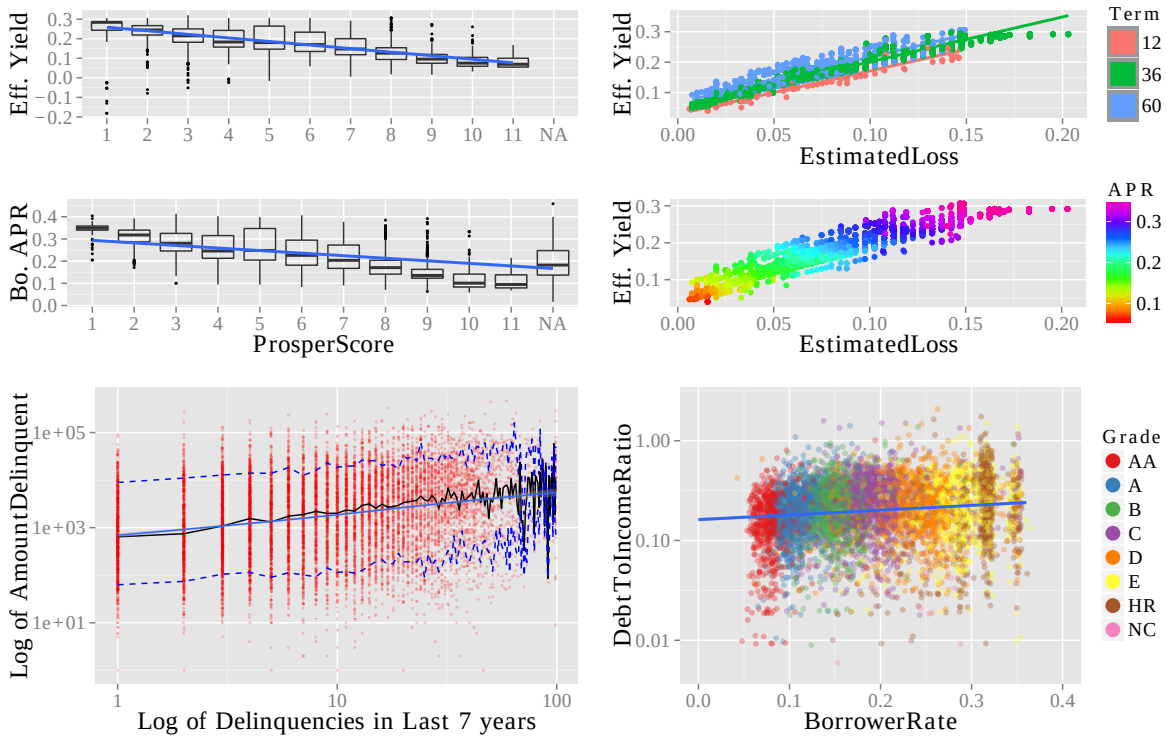


Figure 53: Figure showing the linear trends in the data. Some of the trends become apparent only when the variables are viewed in log-scale. Understanding such trends allows one to pick features that affect the outcome of any predictive model that might be constructed.

- Most of the loans have a constant service fee of 0.01
- California takes the most loans and Wyoming has the least.
- In Fig. 50 we show that all the loans that are more than 60 months since origination are the loans that are completed,charged-off, defaulted or cancelled.

## Reflection:

Many pieces of information are brought together to get an idea of trends in the data and to understand some of the features playing an important role in shaping these trends. One aim of undertaking such an exploratory analysis of the data would be to build models that can help decide the BorrowerAPR and other terms of a loan when a new Borrower applies for a loan, given his current "feature" values. By knowing which variable have a discernible effect on the outcome, it becomes easier to make a more accurate model. As a Borrower it is useful to understand what features are most valued by the Lender so that he can position himself better to get a better rate on the loan.

The lack of a clear objective when I got the data was a difficulty early on in the analysis. Matrixplots were too dense to make sense at first and required breaking them up. In the above analysis some conflict between dplyr and plyr (specifically the order in which they are loaded) caused some errors in compiling the KnitPDF document to be submitted. Other challenges were in the data itself: while it seemed that the linear zones in the data (Fig.33) was based on Loan Status, it revealed that only part of the data (Fig. 34) could be explained by that kind of splitting along LoanStatus (Done vs On-Going).

I went through the list of variables and their description in the excel sheet and selected variables that seemed to affect the loan data and narrow down the list of relevant variables. It would be great if the variables could

be broken to k-buckets and the ggpairs package could generate $k^2$ subplots to look at the dependence of the variables (k=2 or 3 at most). I tried to break the matrix plot in to subplots and catch any interesting correlations in the subplots.

Another problem was applying varying color (i.e. colorbar instead of legend in *aes* of the plot) to each bar in a histogram (as in Fig.48, where returns are broken up in to intervals instead of just being a continuous variable). It was impossible to apply a varying color (like coloring the histogram of returns by a continuous variable like EstimatedReturn instead of breaking it to buckets using the *cut* command). While aes(color=*as.factor*(CreditScoreRangeLower)) achieves the result it could not provide a good legend. This was solved by breaking the continuous variable in to buckets and coloring by buckets.

Writing the relationship between some of the crucial variables made it slightly easier to see which variables to focus on: for example, knowing that Returns and Yield were linearly related made it obvious that they would be correlated. However, knowing that Yield and Loss have no explicit relation between them and are seemingly unrelated made the finding of linearity in the Yield vs Loss plot very interesting.

Delineating data based on Term was important to get the linear zones in many of the plots demarcated. However for Term ==36, the linear zones was not easy to demarcate. A further splitting along 'LoanMonthsSinceOrigination' demarcated the linear and nonlinear zones of 36 month loans. Through trial and error it was found that LoanMonthsSinceOrigination ==39 was a crucial threshold that separated to regimes in the data. The 36 month loan which had LoanMonthsSinceOrigination >=39 was the only regime where the Lender had negative returns.

As mentioned earlier, one main objective of such an exploration and understanding the trends in the data would be develop a model that can, for example, calculate the APR for a loan once the borrower's details are known. This would be achieved by employing machine learning techniques on the existing data and using a classification algorithm to determine if the borrower is risky or not, or using regression to determine the APR for a given application.